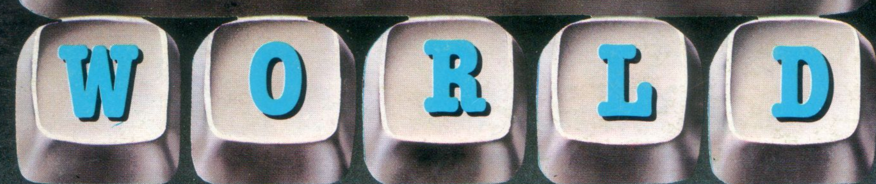
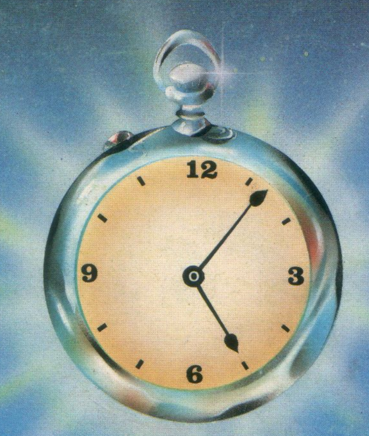


AMIGA
WORLD
Sección fija

N.º 61 - JULIO-AGOSTO 1989 - Publicación de IDG Communications - 400 ptas.



CRONOMETRO
Código Máquina

KALAH
Juego Africano

RUTINAS 128



ISABEL TAPIAS

COMMODORE AMIGA WORLD

Desde enero de 1990 AMIGA WORLD tendrá periodicidad mensual.
SUSCRIBASE AHORA Y PAGUE EN ENERO DEL 90



VENTAJAS QUE OBTIENE AL SUSCRIBIRSE AHORA

AHORRO de 825 pts. 15 % de descuento sobre el precio de portada, 11 ejemplares. Usted sólo paga 4.675 pts.



el ejemplar de AMIGA WORLD correspondiente a noviembre 89

BOLETIN DE SUSCRIPCION AMIGA WORLD

Para suscribirse ahora o solicitar más información, sólo es necesario enviar este boletín a: CW Communications (Amiga World), Rafael Calvo, 18, 4.º B, 28010 Madrid.

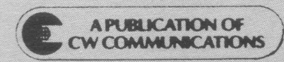
Nombre .... Empresa ....
Dirección .... Población ....
Provincia .... C.P. .... Teléfono ....
Modelo de ordenador y configuración ....
Marque los recuadros con una «X».

- ☐ Suscripción ahora, pagando en enero de 1990 .... 4.675
- ☐ Pago anticipado ahora (ahorro de un 10 % adicional) 4.125
- ☐ Suscripción revista + disco (11 revistas + 11 discos) 18.000
- ☐ Deseo recibir más información sobre Amiga World.

Forma de pago: Cheque, giro o tarjeta de crédito. FIRMA ....
☐ Cheque ☐ Giro (indicar n.º) .... importe ....
☐ VISA ☐ MASTERCARD Número de tarjeta: ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐
FECHA DE CADUCIDAD ....

Enviar a: CW COMMUNICATIONS AMIGA WORLD. Rafael Calvo, 18, 4B. 28010-MADRID

IMPORTANTE: Además de las ventajas que le supone suscribirse ahora a la revista AMIGA WORLD, si prefiere hacernos efectivo el importe por anticipado, tendrá un descuento adicional de un 10 %, con lo que el precio total de la suscripción sería de 4.125 pts., un ahorro de 1.375 pts.



Director General:
Francisco Zabala



Commodore World
está publicado por
CW COMMUNICATIONS, S.A.
y la colaboración
de todos nuestros lectores.

Director:
Juan Manuel Urraca

Dpto. publicidad:
Gloria Montalvo (Madrid)
Magda Zabala (Barcelona)

Redacción Dpto. Técnico:
Alvaro Ibáñez
Aitor Urraca

Diseño:
Miguel Angel Hermosell

Distribución y Suscripciones:
Fernando Rodríguez (dirección),
Angel Rodríguez,
Basilio Rodríguez
Tel.: 419 40 14

COMMODORE WORLD
c/ Rafael Calvo, 18-4.º B
28010 Madrid
Tel. (91) 419 40 14
Télex: 45522
(indicar CW COMMUNICATIONS)
Fax: 419 61 04

DELEGACION EN BARCELONA:
c/ Bertrán, 18-20, 3.º - 4.ª
08023 Barcelona
Tels. (93) 212 73 45/212 88 48

FAX (93) 418 93 55

C.I.F. A.-28-735389

El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo, es de 400 ptas., sin I.V.A.

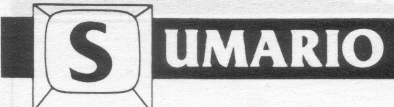
Distribuidora: SGEL
Avda. Valdelaparra, s/n.
Polg. Ind. de Alcobendas - Madrid

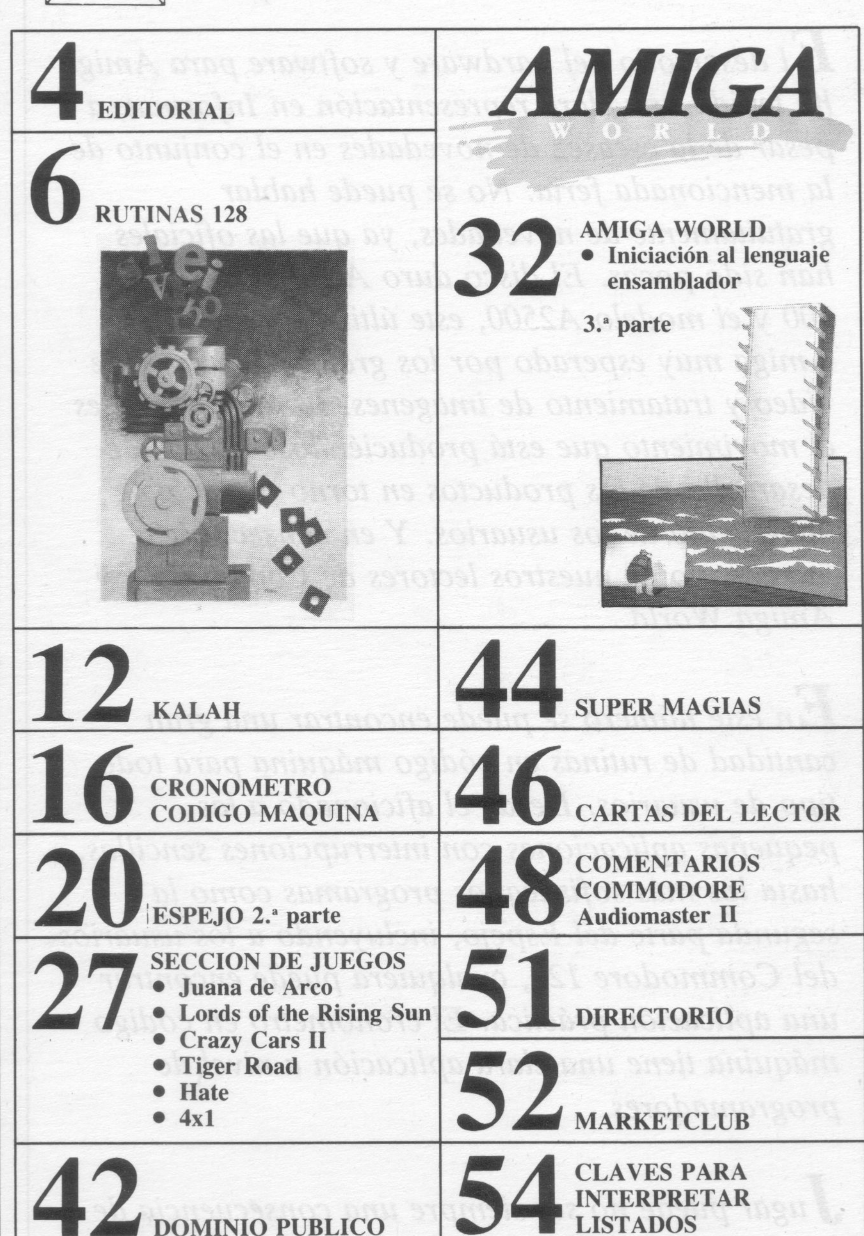
Commodore World
es una publicación
IDG COMMUNICATIONS

control 

Imprime: OMNIA I.G.
Mantuano, 27 - 28002 Madrid
Fotocomposición:
ANDUEZA, S. A.
Fotomecánica: RODACOLOR

Depósito Legal: M-2944-1984





4 EDITORIAL

6 RUTINAS 128

12 KALAH

16 CRONOMETRO CODIGO MAQUINA

20 ESPEJO 2.ª parte

27 SECCION DE JUEGOS
- Juana de Arco
- Lords of the Rising Sun
- Crazy Cars II
- Tiger Road
- Hate
- 4x1

42 DOMINIO PUBLICO

32 AMIGA WORLD
- Iniciación al lenguaje ensamblador

3.ª parte

44 SUPER MAGIAS

46 CARTAS DEL LECTOR

48 COMENTARIOS COMMODORE
Audiomaster II

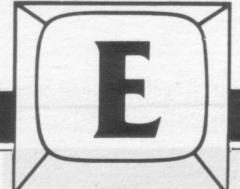
51 DIRECTORIO

52 MARKETCLUB

54 CLAVES PARA INTERPRETAR LISTADOS



COMMODORE WORLD es una publicación de IDG Communications, el mayor grupo editorial del mundo en el ámbito informático. IDG Communications edita más de 90 publicaciones relacionadas con la informática en más de 34 países. Catorce millones de personas leen una o más de estas publicaciones cada mes. Las publicaciones del grupo contribuyen al Servicio Internacional de Noticias de IDG que ofrece las últimas noticias nacionales e internacionales sobre el mundo de la informática. Las publicaciones de IDG Communications incluyen: **ARABIA SAUDI:** Arabian Computer News. **ARGENTINA:** Computerworld Argentina. **ASIA:** Communications World; Computerworld Hong Kong; Computerworld South East Asia; PC Review. **AUSTRALIA:** Computerworld Australia; Communications World; Australian PC World; Australian Macworld. **AUSTRIA:** Computerwelt Oesterreich. **BRASIL:** DataNews; PC Mundo; Micro Mundo. **CANADA:** Computer Data. **CHILE:** Informática; Computacion Personal. **COREA DEL SUR:** Computerworld Korea; PC World Korea. **DINAMARCA:** Computerworld Danmark; PC World Danmark; CAD/CAM World. **ESPAÑA:** Computerworld España; PC World España; Commodore World; Comunicaciones World; CIM World. **ESTADOS UNIDOS:** Amiga World; CD-ROM Review; CIO; Computer Currents; Computerworld; Digital News; Federal Computer Week; 80 Micro; FOCUS Publications; InCider; Infoworld; Macintosh Today; Macworld; Computer + Software News (Micro Marketworld/Lebhar-Friedman); Network World; PC Letter; PC World; Portable Computer Review; Publish!; PC Resource; Run. **FINLANDIA:** Mikro; Tietiviikko. **FRANCIA:** Le Monde Informatique; Distributique; InfoPC; Telecoms International. **GRECIA:** Computer Age. **HOLANDA:** Computerworld Netherlands; PC World Benelux. **HUNGRIA:** Computerworld SZT; PC Microvilag. **INDIA:** Dataquest; PC World India. **ISRAEL:** People & Computers Weekly; SBM Monthly. **ITALIA:** Computerworld Italia. **JAPON:** Computerworld Japan; Semicon News. **MEXICO:** Computerworld Mexico; PC Journal. **NORUEGA:** Computerworld Norge; PC World Norge. **NUEVA ZELANDA:** Computerworld New Zealand. **REINO UNIDO:** Computer News; ICL Today; LOTUS; PC Business World. **REPUBLICA FEDERAL ALEMANA:** Computerwoche; Information Management; PC Welt; PCWoche; Run/Run Specials. **REPUBLICA POPULAR CHINA:** China Computerworld; China Computerworld Monthly. **SUECIA:** Computer Sweden; Mirkro Datorn; Svenska PC World. **SUIZA:** Computerworld Schweiz. **VENEZUELA:** Computerworld Venezuela.



DITORIAL

El desarrollo del hardware y software para Amiga ha tenido una clara representación en Informat, a pesar de la escasez de novedades en el conjunto de la mencionada feria. No se puede hablar gratuitamente de novedades, ya que las oficiales han sido pocas. El disco duro A590 para Amiga 500 y el modelo A2500, este último es un super Amiga muy esperado por los grandes usuarios de vídeo y tratamiento de imágenes. Lo importante es el movimiento que está produciéndose a nivel de desarrollo de los productos en torno al Amiga e incremento de los usuarios. Y en consecuencia, incremento de nuestros lectores de Commodore y Amiga World.

En este número se puede encontrar una gran cantidad de rutinas en código máquina para todo tipo de usuarios. Desde el aficionado a las pequeñas aplicaciones con interrupciones sencillas, hasta los más sofisticados programas como la segunda parte del Espejo, incluyendo a los usuarios del Commodore 128, cualquiera puede encontrar una aplicación práctica. El cronómetro en código máquina tiene una clara aplicación a nivel de programadores.

Jugar puede no ser siempre una consecuencia de la utilización ociosa del ordenador. Kalah, un juego africano bastante sencillo pero curioso, puede servir a los lectores menos experimentados para ahondar en las maravillas del Basic.

En este mismo mes vuelve a aparecer Amiga World. El número tres, que estará dedicado al complejo mundo del vídeo, estará disponible aproximadamente a partir del día 15 de julio. Como siempre, editaremos dos estupendos discos con programas, pantallas y demostraciones.

NOTICIAS

INFORMAT 89

En la última edición de INFORMAT, celebrada entre los días 5 y 10 de junio, se agruparon más de 310 empresas expositoras y fueron representadas cerca de 500, tanto nacionales como extranjeras. El conjunto de la superficie ocupada era de 18.500 metros cuadrados. Estas cifras representan un importante incremento (42,3% en el caso de la superficie ocupada y un 20% en el de expositores) en relación a la edición anterior.

En cuanto a la cifra de visitantes que acudieron al salón, se estima en más de 40.000 profesionales nacionales y extranjeros. En su edición de 1988, INFORMAT recibió la asistencia de 39.380 visitantes pofesionales.

La noticia más importante para nuestra compañía se refiere a la gran difusión de Amiga World. El primer día de la feria se agotó el número dos. El segundo día se agotaron los discos de Amiga y prácticamente también los números 0 y 1. A pesar de reponer ejemplares, continuó esta tónica durante el resto de la feria. En conclusión, un gran acontecimiento para esta publicación que cada vez cuenta con más lectores y suscriptores. ■

NOVEDADES COMMODORE, S. A., PARA INFORMAT

Productos: Disco duro A-590 para Amiga 500, PC 30 III Turbo a 12 MHz, PC 40 III AT a 6/8/12 MHz incluye adaptador integrado para gráficos VGA, PCs 60-40 y 60-80 con CPU 386 de Intel de 32 bits y frecuencia 8/16 MHz, junto a los mencionados productos tuvimos la oportunidad de ver trabajando un Amiga 2500, aunque casi extraoficialmente, bajo sistema Unix. También una pantalla sensible al tacto, conectada a uno de los Amiga que funcionaban enchufados al 2500.

Por otra parte se produjo una pequeña revolución de los Genlocks. Se presentaron novedades como el Super-Pic distribuido por Barnacomputer o el

Videocomp VCG 3 distribuido en nuestro país por Iberunión.

La representación de Commodore compuesta principalmente por sus propios distribuidores, estaba inspirada en el monumental stand que Commodore Alemania disponía en la feria de Hannover. Dos plataformas octogonales donde concurrían importantes distribuidores de la marca y donde se presentaban tanto los Amiga como los PC's y algunos periféricos. Los últimos días de la feria (viernes y sábado) se presentó un «show» con animación y sonido en tiempo real con Amiga, actores reales en el estand y mucho colorido. ■

NUEVA VERSION DEL SISTEMA OPERATIVO DE AMIGA

Commodore anuncia la disponibilidad de una nueva versión del Sistema Operativo WORKBENCH del Amiga. Esta aplicación de software en su nueva versión 1.3, se reconoce como producto Amiga A-5211.

Incluye tres discos:

- Disco de kickstart versión 1.3 (para usuarios de Amiga 1000).
- Disco Workbench versión 1.3.
- Disco de Extras.

Y el correspondiente manual en castellano.

La versión 1.3 mejora la tecnología existente, amplía las capacidades del Amiga y refuerza muchas aplicaciones de software.

Se han añadido varios programas y comandos nuevos. Por ejemplo, el programa SHELL que permite editar líneas y revisar comandos ejecutados con anterioridad, o el FIXFONTS Workbench después de añadir nuevos tipos. MORE posibilita visualizar ficheros de texto ASCII y CMD redirige la salida serie o paralelo a un fichero.

La característica RAD permite definir la memoria central como un disco, lo que proporciona un rápido acceso a los datos. Además, se han aumentado las capacidades de los comandos originales y toda la información completa se encuentra en el manual que va con esta nueva versión 1.3.

En el disco de Extras hay directorios con nuevos drivers de impresoras que permiten conectar el equipo con, prácticamente, todas las impresoras del mercado. ■

COMMODORE CONSTRUYE UNA NUEVA PLANTA DE ALTA TECNOLOGIA EN ALEMANIA

Commodore Internacional va a sustituir su actual planta de fabricación de Braunschweing, Alemania, por unas modernas instalaciones diseñadas para incrementar la eficiencia y flexibilidad de producción de la compañía. El nuevo edificio, que estará también localizado en Braunschweing, quedará terminado el año próximo y tendrá capacidad de producir más de un millón de ordenadores al año.

Estas modernas instalaciones representan un importante paso estratégico para Commodore, porque aumenta la capacidad de producción en Alemania, donde tienen un gran mercado. A su vez, este incremento de la capacidad de producción ayudará a la compañía a soportar sus objetivos de crecimiento en Europa y a solidificar su ya fuerte posición en el mercado de ordenadores personales, PC's, donde ocupa el segundo lugar por número de unidades entregadas en 1988.

La planta está diseñada para producir una extensa gama de equipos Commodore, Amiga y PC's, así como otros nuevos productos que están actualmente en desarrollo. Representa un paso más dentro de los que COMMODORE está dando para intensificar sus operaciones a nivel mundial con crecimientos importantes en la venta de sus ordenadores personales y compatibles. ■

NUEVO COMMODORE PC 30-III



Madrid, 1 de junio de 1989. Commodore anuncia la disponibilidad en España, al mismo tiempo que en Europa, del nuevo PC Compatible AT modelo PC 30-III. Con este equipo, la Compañía aporta una revolucionaria relación precio/características, ya que es un ordenador de gran rendimiento a un precio bajo y con calidad de fabricación alemana.

El PC 30-III incluye como características standard el Procesador 80286 Turbo a 12 MHz, 640 Kb de memoria RAM ampliables hasta 16 Mb, disco duro de 20 Mb, floppy de 3,5" y 1.44 Mb de capacidad, interfaces serie, paralelo, vídeo y ratón. Además, dispone de 4 slots de ampliación, dispone de reloj/calendario de tiempo real, alimentado por batería de carga automática.

El usuario puede elegir entre tres velocidades de reloj (6, 8 y 12 MHz-Turbo) para asegurar el máximo rendimiento y compatibilidad con una amplia variedad de aplicaciones de software.

En su configuración básica se incluye el teclado DIN de 102 teclas y un monitor monocromo de 14", ergonómico y ámbar. Además de gráficos EGA, facilita también gráficos MDA, CGA y Hércules.

Una característica de este equipo es la Autoconfiguración BIOS desarrollada por COMMODORE que reconoce de manera totalmente automática las ampliaciones y configura el sistema de acuerdo con ellas.

En la configuración básica incorpora un disco duro de 20 millones de caracteres, con una gran velocidad de transferencia y proceso, gracias también al acelerador de transmisión de datos a través del BUS del procesador. Este disco puede reemplazarse o ampliarse con unidades de 40, 80, 120 Mb. ■

RUTINAS 64/128

Por Rafael López Rubio

Las rutinas que a continuación presentamos, se pueden utilizar con el C-128 ya que están diseñadas especialmente para ese modelo de Commodore. Pero también se incluye algunas que funcionan en el C-64 sin más problemas que algún POKE. Lo importante es que son programas de utilidad prácticas.

KEYS 9-10

El comando KEY no permite definir las teclas de función HELP y SHIFT-STOP. Para conseguirlo hay que usar directamente la rutina del kernal. Esta, necesita un puntero en la página cero que puede obtenerse a través de los punteros de strings temporales (27-35). Se puede transformar un string en temporal añadiendo un CHR$() al final del mismo.

Hay que enviar tres parámetros después del SYS, el puntero, el número de tecla de función y la longitud del string.

El programa KEYS DEMO modifica la función SHIFT-STOP para poder ejecutar un fichero a partir del directorio. La tecla HELP se ha definido de una manera instructiva.

```
PROGRAMA: KEYS.9-10

1 REM COMMODORE WORLD 61                 .215
2 :                                      .234
3 REM RAFAEL LOPEZ RUBIO                 .53
5 :                                      .237
10 A$="ESTO ES SHIFT-STOP"+CHR$(0):      .180
K=9:GOSUB40
20 A$="ESTO ES HELP"+CHR$(0):K=10:G     .60
OSUB40
30 END                                   .32
40 BANK15:SYS65381,28,K,LEN(A$):RET     .108
URN

PROGRAMA: KEYS.DEMO

1 REM COMMODORE WORLD 61                 .215
2 :                                      .234
3 REM RAFAEL LOPEZ RUBIO                 .53
5 :                                      .237
10 A$="RUN"+CHR$(27)+"K"+CHR$(20)+C     .238
HR$(20)+CHR$(20)+CHR$(27)+"@"+CHR$(
13)
20 BANK15:SYS65381,28,9,LEN(A$)         .220
30 :                                     .6
40 B$="SCNCLR:BANK15:FORI=17431TO18     .16
168:A=PEEK(I):IFA>90THENA=A-128:?CH
R$(A)CHR$(32);:NEXT:ELSE:?CHR$(A);:
NEXT"+CHR$(13)
50 BANK15:SYS65381,28,10,LEN(B$)        .4
```

Algunos registros del VDC producen efectos curiosos, y otros tienen una unidad concreta.

VDC TRAINER

Una buena manera de enfrentarse a algunos chips difíciles de manejar consiste en hacerse un programa entrenador. Mediante VDC TRAINER podrás descubrir el chip de vídeo de 80 columnas VDC, familiarizarte con sus registros y adquirir la experiencia que te permita manipularlo con facilidad.

Cómo usarlo:

Debes echar mano del antiguo televisor y conectarlo al C-128. Ejecuta VDC TRAINER en la pantalla de 40 columnas mientras observas los resultados en el monitor.

El chip VDC se controla por 37 registros que se muestran en la pantalla. El extraño cursor que también aparece permite incrementar o decrementar el valor de estos registros al pulsar la barra de espacio, dependiendo del sentido que señale la flecha. Para escribir en un registro sin modificarlo colócate en los espacios entre números.

Algunos registros producen efectos curiosos, otros tienen una utilidad concreta y los demás se emplean conjuntamente.

Las instrucciones están en el mismo listado. Si obtienes unos valores interesantes puedes guardarlos en la línea 1020 con la función D del menú. El programa lleva incluida una línea 1020 demo que simula un monitor monocromo haciendo GOSUB 1000. Dispones de la línea 530 para insertar tu propia rutina de prueba que se ejecuta con la función U del menú.

Registros de uso conjunto:

Sin embargo algunos registros tienen utilidad difícil de ver porque se usan en combinación. El VDC se sirve de sus propios 16K de RAM (VRAM) para formar la pantalla, de tal manera que nuestro C-128 viene a ser un C-144.

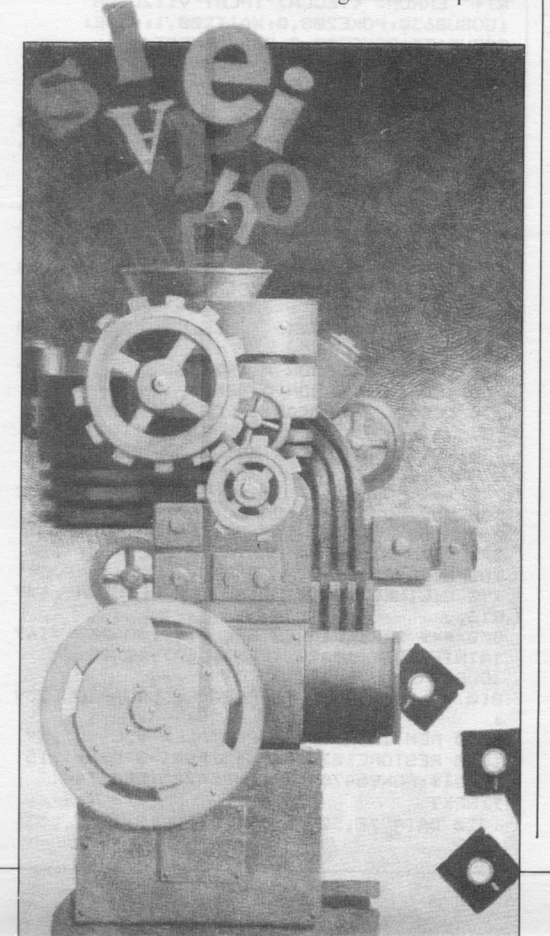
No se puede acceder a un registro de la

VRAM (rVRM) con PEEK y POKE, sino que debe hacerse a través de los registros del VDC (rVDC), que son precisamente los que aparecen en la pantalla de este programa. Pero tampoco con PEEK ni POKE se controla los rVDC directamente. Se hace a través de las direcciones $D600 y $D601 de las E/S, que sí que se manejan con PEEK-POKE. Estas posiciones de memoria se denominan ports.

En $D600 se POKEa el rVDC (0-36) y después en $D601 se lee su contenido o se escribe un valor mediante PEEK o POKE (En C.M. usar el kernal en $CDCC y $CDDA). Los rVDC de uso conjunto son:

— Leer un rVRAM: Escribir la dirección del rRAM en los rVDC 18-19. Leer el rVDC 31.

— Leer varios rVRAM seguidos: El primero se lee como en el punto anterior. Los demás basta con ir leyendo el rVDC 31. Los rVDC 18-19 aumentan en 1 cada vez que se accede al rVDC 31.



— Escribir en uno o varios rVRAM seguidos: Igual que en los dos puntos previos pero escribiendo en el rVDC 31. Esto no funciona en BASIC. No obstante el programa usa una rutina del kernal en lugar de los dos POKEs con lo que se solventa el problema.

— Rellenar una zona de la VRAM: Colocar a cero el bit 7 del rVDC 24. Escribir la dirección de inicio en los rVDC 18-19 Colocar en el rVDC 31 el byte de relleno (al colocarlo se escribe en la VRAM, excepto desde el BASIC). Escribir en el rVDC 30 el número de bytes a rellenar menos uno (menos el que ya se ha escrito), si se escribe un cero se rellenan 256 bytes. Si se desea rellenar un área mayor, continuar escribiendo en el rVDC 30 hasta completar el área.

— Mover una zona dentro de la VRAM: Colocar a uno el bit 7 del rVDC 24. Escribir en los rVDC 32-33 la dirección de origen de la VRAM a copiar. Escribir en los rVDC 18-19 la dirección de destino. Escribir en el rVDC 30 el número de bytes a trasladar, si se escribe un cero se trasladan 256 bytes. Si se desea trasladar un área mayor, continuar escribiendo en el rVDC 30 hasta completar el área.

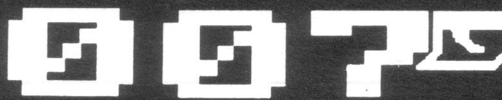
Una buena manera de enfrentarse con los chips difíciles de manejar consisten en emplear un programa entrenador.

```
PROGRAMA: VDC.TRAINER

1 REM COMMODORE WORLD 61                .215
2 :                                     .234
3 REM RAFAEL LOPEZ RUBIO                .53
5 :                                     .237
100 REM *** INICIALIZACION VDC TRAI     .242
NER
110 REM * PANTALLA                      .0
120 FAST:GRAPHICCLR:GRAPHIC0,1:FORI     .62
=0TO5:READJ:COLORI,J:NEXT:DATA16,1,
2,15,16,1:PRINTCHR$(11)CHR$(142):SL
OW
130 REM * VARIOS                        .206
140 BANK15:SYS65381,0,9,0:POKE828,1     .100
83:TRAP610
150 REM * VARIABLES (GENERAL=I,J,K,     .124
L,E,X,Y,I$)
160 AD=54784:DA=AD+1:DIMVDC(36),HLP     .32
$(38),INC(15)
170 XX=4:YY=1:C1=2:C2=3:HL=0:MH=HL:     .166
DT$=CHR$(20)+CHR$(13)+"SCNCLR:SPRIT
```

El programa KEYS 9-10 permite redefinir las teclas HELP y SHIFT-RUN/STOP en el C-128.

```
E1,0:SPRITE2,0:POKE828,173:LIST1000
-"
180 FORI=1TO15:READJ:INC(I)=J:NEXT:  .86
DATA128,64,32,16,8,4,2,1,,100,10,1,
,16,1
190 FORI=0TO37:READI$:HLP$(I)=I$:PR .10
INTI;I$:NEXT:SCNCLR
200 DATA $00 HORIZONTAL TOTAL,$01 H .208
ORIZONTAL DISPLAYED,$02 HORIZONTAL
SYNCRO POSITION,$03 7-4 VERTICAL WI
DTH/3-0 HORIZONTAL   WIDTH,$04 VERT
ICAL TOTAL
210 DATA $05 7-5 XXX/4-0 VERTICAL T .80
OTAL ADJUST,$06 VERTICAL DISPLAYED,
$07 VERTICAL SYNCRO POSITION,$08 7-
2 XXX/1-0 INTERLACE MODE
220 DATA $09 7-5 XXX/4-0 CHARACTER  .138
VERTICAL,$0A 7 XXX/6-5 CURSOR MODE/
4-0 CURSOR    START LINE,$0B 7-5 XX
X/4-0 CURSOR END LINE,$0C DISPLAY A
DDRESS HIGH
230 DATA $0D DISPLAY ADDRESS LOW,$0 .118
E CURSOR POSITION HIGH,$0F CURSOR P
OSITION LOW,$10 LIGHT PEN VERTICAL
(READ),$11 LIGHT PEN HORIZONTAL (RE
AD)
240 DATA $12 UPDATE ADDRESS HIGH,$1 .100
3 UPDATE ADDRESS LOW,$14 ATTRIBUTE
ADDRESS HIGH,$15 ATTRIBUTE ADDRESS
LOW
250 DATA $16 7-4 CHAR. HORIZONTAL T .90
OTAL/3-0 CHAR. WIDTH,$17 CHAR. LENG
TH,$18 7 UPDATE_BLOCK(1)-UPDATE_DAT
A/6    REVERSE MODE/5 BLINK RATE/4
-0 VERTICAL  SCROLL
260 DATA $19 7 HI-RES/6 MONOCHROME/ .221
5 SEMIGRAPHIC/4 DOUBLE WIDTH/3-0 HO
RIZONTAL SCROLL
270 DATA $1A 7-4 FOREGROUND MONOCHR .119
OME COLOR/3-0 BACKGROUND COLOR,$1B
ADDRESS INCREMENT ROW,$1C 7-5 CHARA
CTER BASE ADDRESS/4 RAM     TYPE/3-0
 XXX
280 DATA $1D 7-5 XXX/4-0 UNDERLINE, .65
$1E WORDCOUNT (COPY BLOCK-FILL UPDA
TE),$1F DATA (FILL-READ UPDATE),$20
 BLOCK COPY ADDRESS HIGH
290 DATA $21 BLOCK COPY ADDRESS LOW .173
,$22 DISPLAY ENABLE BEGIN,$23 DISPL
AY ENABLE END,$24 7-4 XXX/3-0 REFRE
SH RATE,"[RVSON]MENU>[RVSOFF] VDCCL
R DATA PRINT SCNCLR USR"
300 REM * SPRITES                   .97
310 FORI=3584TO3711:POKEI,0:NEXT:SP .49
RITE1,1,2:SPRITE2,1,,,,,1:FORI=1TO2
:MOVSPRI,55,53:NEXT:FORI=3648TO3660
STEP3:POKEI,85:POKEI+39,255:NEXT
320 J=0:FORI=51TO3STEP-6:FORJ=JTOJ+ .25
2:READI$:K=DEC(I$):POKE3584+J,K:POK
E3584+J+I,K:NEXTJ,I:DATA FF,C0,,F3,
C0,,E1,C0,,C0,C0,,FF,C0,,80,40,,80,
40,,80,40,,80,40,
330 REM *** DESARROLLO DEL PROGRAMA .211
340 GOSUB560:HL=0:GOSUB630:POKE208, .225
0
350 DO:SPRCOLORC1,C2:GETKEYI$:ONINS .1
TR("[CRSRU][CRSRR][CRSRD][CRSRL] [F
LCH ARRIBA][HOM][CLR]"+CHR$(13)+CHR
$(141),I$)GOTO360,370,380,390,400,4
10,420,430,440,440:PLAY"V1T204SA":L
OOP
360 IFC1=3THENX=0:Y=8*(YY>1):HL=HL+ .169
Y/8:GOSUB650:LOOP:ELSEC1=3:C2=2:LOO
P
370 I=XX=18:Y=0:X=8*-(XX<38)-40*I:H .167
L=HL-19*I:GOSUB650:LOOP
380 IFC2=3THENX=0:Y=8*-(YY<19):HL=H .5
L+Y/8:GOSUB650:LOOP:ELSEC2=3:C1=2:L
OOP
390 I=XX=24:Y=0:X=8*(XX>4)+40*I:HL= .207
HL+19*I:GOSUB650:LOOP
400 IFHL=37THEN450:ELSEVDC(HL)=255A .155
ND(VDC(HL)+INC(XX-3+20*(XX>23))*(C1
-C2)):GOSUB580:SYSDEC("CDCC"),VDC(H
L),HL:LOOP
410 IFHL=37THEN460:ELSEPOKEAD,HL:VD .91
C(HL)=PEEK(DA):GOSUB580:LOOP
420 XX=4:YY=1:HL=0:FORI=1TO2:MOVSPR .117
I,55,53:NEXT:GOSUB630:LOOP
430 XX=24:YY=19:HL=37:FORI=1TO2:MOV .57
SPRI,215,197:NEXT:GOSUB630:LOOP
440 IFHL=37THEN460:ELSEI=480-540*(I .41
$=CHR$(141)):RESTOREI:FORI=0TOHL:RE
ADI$:NEXT:J=DEC(I$):VDC(HL)=J:SYSDE
C("CDCC"),J,HL:GOSUB580:LOOP:*RESTO
RE VARIABLE!!!!!!!*
450 ONXX-23GOTO470,460,490,460,500, .81
460,520,460,530:REM MENU V D P S U
460 PLAY"V1T204SA":LOOP             .225
470 RESTORE480:FORI=0TO36:READI$:PO .161
KEAD,I:POKEDA,DEC(I$):NEXT:GOSUB560
:LOOP
480 DATA 7E,50,66,49,27,E0,19,20,FC .121
,E7,A0,E7,,,,,ED,34,,,8,,78,E8,20,4
7,F0,,2F,E7,1,1,,,7D,64,F5
490 PRINT"[CLR]1020DATA ";:FORI=0TO .169
36:PRINTMID$(HEX$(VDC(I)),3)",";:NE
XT:PRINTDT$:POKE208,3:POKE842,19:PO
KE843,13:POKE844,13:END
500 RESTORE500:GRAPHIC5:PRINT"[CYN] .217
":FORI=1TO11:READJ:FORK=0TO4:PRINTC
HR$(J)"VDC TRAINER. PROGRAM",:NEXT:
PRINT:NEXT:DATA142,2,15,14,5,28,30,
129,149,151,5
510 POKE241,PEEK(241)OR64:FORI=1TO4 .25
:PRINT"VDC TRAINER. PROGRAM",:NEXT:
POKE241,PEEK(241)AND191:GRAPHIC0:LO
OP
520 GRAPHIC5,1:GRAPHIC0:LOOP        .110
530 :LOOP:REM LINEA USUARIO         .124
540 *** SUBRUTINAS                  .90
550 * INICIALIZA                    .140
560 PRINT"[HOM]"TAB(16)"VDC[2SPC]TR .218
AINER":FORHL=0TO36:POKEAD,HL:VDC(HL
)=PEEK(DA):GOSUB580:NEXT:PRINTTAB(2
4)"V D P S U":FORI=0TO39:PRINT"[SHI
FT*]";:NEXT:RETURN
570 * IMPRIME UN VALOR VDC          .92
580 I=HL:L=VDC(I):CHAR,1-20*(I>18), .114
I+1+19*(I>18):PRINTUSING"##:";I;:J=
128:FORK=0TO7:PRINTCHR$(48-((LANDJ)
=J));:J=J/2:NEXT
590 PRINTUSING".###";L/1000;:PRINT" .88
[4CRSRL] [3CRSRR] "MID$(HEX$(L),3):
I=FRE(1):RETURN
600 * ERROR TRAP                    .190
610 SLOW:E=HL:HL=38:HLP$(38)=ERR$(E .190
R)+" ERROR. <TECLA>":PLAY"V1T206SA"
:GOSUB630:POKE208,0:WAIT208,1:HL=E:
GOSUB630:POKE208,0:RESUMENEXT
620 * IMPRIME PIE                   .56
630 CHAR,0,21:PRINTCHR$(27)"@";HLP$ .30
(HL):RETURN
640 * MUEVE CURSOR                  .178
650 MOVSPR1,+X,+Y:MOVSPR2,+X,+Y:XX= .126
XX+X/8:YY=YY+Y/8:IFMH<>HLTHENMH=HL:
GOTO630:ELSERETURN
660 *** INSTRUCCIONES:              .206
670 <CRSRS>:MOVER CURSOR Y SU FLECH .184
A.
680 <SPC>  :AUMENTAR O DISMINUIR NU .108
MERO SEGUN FLECHA Y ESCRIBIR REGIST
RO.
690 <[FLCH ARRIBA]>     :LEER REGIST .180
RO.
700 <HOME> :CURSOR A INICIO.        .112
710 <CLR>  :CURSOR A MENU.          .4
720 <CR>   :COLOCAR VALOR POR DEFEC .56
TO.
730 SH-<CR>:COLOCAR EL VALOR DE LA  .70
LINEA 1020.
740 *** MENU:                       .182
750 V:VALORES POR DEFECTO DEL VDC.  .66
760 D:OBTENER LINEAS DATA USUARIO.  .84
770 P:IMPRIMIR EN PANTALLA 80C.     .19
780 S:BORRAR PANTALLA 80C.          .29
790 U:LINEA 530 PARA RUTINA DE USUA .177
RIO.
800 *** DIRECCIONES DEL O.S.:$0A2F  .167
(ATRIBUTOS);$0A2E (PANTALLA);$0A2B
(CURSOR)
810 *** LINEAS DATA PARA EL USUARIO .239
:
1000 REM INICIALIZA PANTALLA VDC    .89
1010 RESTORE1020:BANK15:FORI=0TO36: .15
READI$:POKE54784,I:POKE54785,DEC(I$
):NEXT
1020 DATA 7E,50,66,49,22,E0,19,1E,F .33
```



ALCALA, 211.
28028 MADRID.
Tel. 91/256 14 30.
Fax 91/256 16 91

DISTRIBUIDOR OFICIAL
COMMODORE



PUNTO OFICIAL DE VENTA
DE AMIGAZETA

Amiga Hardware

- Expans. 8M A2000 -395.000
- Expans. 2.M A2000(Ampl.8 M)... -98.000
- Expans. 1/2 M - 33.000
- Expans. 2M A500 -140.000
- Modulador T.V A-500 - 6.500
- Placa Video Comp. A-2000 -15.000
- Placa XT +Unidad 5 1/4 - 95.000
- Placa AT (1 Mega de memoria).. -185.000
- Digitalizador Stereo Audio - 20.000
- Perfect Sound Digit. - 25.000
- Prosound Designer Digit. - 25.000
- Future Sound Digitalizador - 32.000
- Digitot (Audio + Video) - 34.500
- Funda Polivinilo A-2000 - 2.500
- Funda Polivinilo A-500 - 1.975
- Archivador 10 diskettes - 300
- Archivador Metalico 70 D - 4.000
- Archivador Plastico 150 D - 3.500
- Alfombrilla Para Raton - 2.000
- Conmutador 2 perif. paralelo - 16.500
- Impresora Color Nec 24 Agujas
- P6 Plus 80 C./264 C.P.S. -170.000
- Impr. Color Star 9 LC-10 - 75.000
- Tableta Grafica Easyl A-2000 .... - 90.000
- Tabl. Grafica Easyl A500/A1000 - 80.000
- Genlock Rendall/Ariadne - 79.000
- Genlock con funcion Invert - 70.000
- Genlock Commodore A-2000 - 49.000
- MiniGen (Genlock) - 39.000
- **DIGI VIEW GOLD 3,0 PAL** - 29.500
- Adaptador D.View A500/A2000 - 5.000
- Digipic(Tiempo real monoc.) - 70.000
- Perfect Vision(T. real monoc.) - 49.000
- Unidad 3 1/2" Ext Commodore - 35.000
- Unidad 3 1/2" Ext No Commod. - 30.000
- Unidad 3 1/2" Int Commodore - 30.000
- Unidad 3 1/2" Int No Commod - 25.000
- Unidad 5 1/4" Ext (80 pistas). - 40.000
- Unidad 5 1/4" Ext (40 pistas) - 35.000
- Disco Duro 40 Megas A-2000 -185.000
- Disco Duro 20 Megas A-2000 -115.000
- Disco Duro 20 Megas A-500 -140.000
- Midi 2 Out - 24.000
- **Leedor De Pistas** - 20.000
- **Proccesor Accelerator** - 35.000
- **FLICKER MASTER (FILTRO)** - 4.500
- Euroconector Amiga-Sony Trin.. - 5.500
- Conector RGB 23 pines - 1.500
- **Separador Color** - 60.000
- **Frame Buffer 2 Megas** 130.000
- **Genlock** - 88.500
- **Disco D. 20 M A-500 Comm** -98.000

Tratamientos de textos

- Wordperfect - 48.000
- Textcraft Plus - 9.500
- Prowrite - 20.000
- Textpro - 14.000
- Kind Words (Español) - 13.500
- Calligrapher - 23.000
- Ashas.Calligra.Fonts - 13.500
- Call. Newsletters Fonts - 8.000
- Call. Studio Fonts - 8.000
- Fancy 3D Fonts - 12.000
- Zuma Fonts Vol. 1.2.3.4 - 7.000
- Aloha Fonst Vol. 1.2.3 - 6.500
- NTF Fonts Vol. 1.2.3 - 14.000
- Aloha Fonts (del 1 al 3) - 6.500

Paquete Commodore – 30.000
(Superbase,Logistix,Musica.......)

Autoedición

- Pagesetter - 25.000
- Professional Page - 60.000
- **PAGESTREAM** - 35.000

Base de Datos

- Superbase Personal - 19.500
- Video Wizard - 12.500
- Data Retrieve - 14.000
- **MICROFICHE FILER PLUS**- 25.000

Comunicaciones

- Aegis Diga! - 13.500

Emuladores

- Dos 2 Dos - 12.500
- Disk 2 Disk - 12.500
- Transformer (pc) - 6500
- 64 Emulator II - 12.000

Gestión

- Analyze 2.0 - 25.000
- Quarterback - 11.000
- Galileo 2.0 - 17.000
- **EZ-BACKUP** - 11.000
- **SUPERBACK** - 11.000

Gráficos / Vídeo

- Printm. Plus Art Gallery 1 - 7.000
- Printm.Plus Art Gallery 2 - 7.000
- Aegis Videoscape 3D - 30.000
- Aegis Videoscape 3D 2.0 - 35.000
- Aegis Impact - 14.000
- Aegis Animator+Aegis Images - 22.000
- Animation Apprentice - 40.000
- Animation Multiplane - 13.000
- Animation Editor - 10.000
- Animation Effects - 9.000
- Animation Stand - 9.000
- Animation Flipper - 9.000
- Pro Video CGI Plus Pal - 56.500
- Pro Video Set Fonts I - 20.000
- Pro Video Set Fonts II - 20.000
- Deluxe Library - 5.000
- Sculpt 3D - 25.000
- Animate 3D - 25.000
- Sculpt 4D - 85.000
- Digi paint - 12.000
- Interchange 3D - 10.500
- Int Forms In Flight Conv. mod. - 5.000
- Interchange Objects Disk 1 - 4.000
- **INTERFONT** - 18.000
- Introcad 2.0 - 12.000
- New Technology Coloring Book - 7.000
- Deluxe Video 1.0 - 10.000
- Deluxe Video 1.2 - 13.500
- Aegis Videotitler - 25.000
- Aegis Modeler 3D - 16.000
- **PHOTON VIDEO C.ANIM** - 35.000
- Graphicraft - 6.500
- Dynamic Cad 2.3 - 65.000
- Forms In Flight II - 16.000
- Express Paint - 18.000
- Deluxe Productions - 39.000
- Deluxe Photolab - 13.500
- Deluxe Paint II Pal - 13.500
- **DELUXE PAINT III** - 25.000
- **PHOTON PAINT 2** - 25.000
- TV Text - 18.000
- TV Show - 18.000
- Lights!.Camera!.Action! - 25.000
- Turbo Silver 3D - 40.000
- Fantavision - 20.000
- Pageflipper - 9.500
- Video Visions Fonts vol.1 y 2 - 6.500
- Deluxe Help Digi Paint - 6.000
- Deluxe Help Photon Paint - 6.000
- Deluxe Help Deluxe II - 6.000
- Video Effects - 45.000
- Invision - 25.000
- Clip Art Magnetics 1 al 8 - 5.000
- Cad Parts - 6.000
- Fine Print - 8.500
- Turbo Print - 8.000
- Desktop Artist 1 - 6.500
- Comic Setter - 14.000
- Comic Setter Science - 7.000
- Comic Setter Heroes - 7.000
- Comic Setter Figures - 7.000
- **ZOETROPE** - 23.500
- Human Design - 7.500
- Macrobot D. - 7.500
- Future D. - 7.500
- Architectural D. - 7.500
- Desktop Artist - 6.500
- **DESIGN 3D** - 18.000
- Moviesetter - 18.000
- **C-LIGHT** - 18.000
- Cad Parts - 6.000

Lenguajes

- Assempro - 18.000
- Lattice DBC III Library - 25.000
- Aztec C - 38.000
- Aztec C SDB(for DBase) - 15.000
- Modula 2 - 38.000
- Library For Modula 2 - 29.500
- Arexx - 9.000
- Metacomco Pascal - 19.000
- Pro Fortran 77 - 40.000
- Multi Forth - 19.000
- Devpac - 18.000
- MacroAssembler - 20.000

Música

- Aegis Sonix 2.0 - 15.000
- Aegis Audiomaster II - 18.000
- Deluxe Music Construction Set - 13.500
- The Music Studio - 13.000
- Pro Sound Designer - 9.000
- Pro Sound Midi Plus - 10.500
- Dr.T's KCS 1.6A - 45.000
- Dr.T's Copyist Professional - 55.000
- Midi Magic - 22.000
- E.C.T. Sampleware (del 1 al 4) - 6.500
- Dynamic Drums - 9.500
- A Drum - 9.500
- **SOUND OASIS** - 14.000

Amiga librería

- Amiga Para Principiantes - 4.134
- 68.000 Guia Del Usuario - 1.900
- First Steps In Assembly Lang - 5.475
- Alphabets - 3.500
- Computer Animations - 5.475
- Amiga Applications - 4.675
- Amiga Handbook - 4.775
- Advanced Amiga Basic - 5.975
- Am. System Programmer's Guide - 5.475
- ROM Kernel Ref.Manual:EXEC.. - 5.475
- ROM K. Ref.Manual:Libraries - 7.975
- Developers Reference Guide - 3.500
- A.System Programmer's Guide - 5.975
- C: Progr. Graph. On Amiga
- And Atari ST - 4.500
- Inside Amiga Graphics - 4.900
- Compute's A.Dos Ref. Guide - 4.900
- 68000 Assembly Language - 5.975
- Amiga Machine Language - 5.475
- Bantam's Amiga Dos Manual - 5.975
- Inside The Amiga With C - 5.975
- Hardware Reference Manual - 5.475
- Becomming an Amiga artist - 5.475
- Amiga for Beguinners - 5.475
- Amiga Basic-Inside and Out - 5.595
- Amiga 3D Graphic in Basic - 5.475
- Amiga Tricks and Tips - 5.475
- Amiga Dos: Inside and Out - 5.475
- Amiga Disk Drives - 5.995
- Amiga C for Beguinners - 5.975
- Using Deluxe Paint II - 5.475

Revistas

- Amiga World(USA) - 830

```
C,E7,C0,FF,,,,,,,F,D0,8,,78,E8,20,7
,40,,2F,E7,4F,7,F,80,7D,64,F5:POKE2
603,64:REM LINEA DEMO
1030 RETURN                        .68
```

ESPECIAL HEX

Esta rutina facilita la copia de los largos programas llenos de DATAs en exadecimal. Para ello convierte el teclado numérico y las cuatro funciones en un teclado hexadecimal. Las teclas F1 a F7 imprimen A, B, C y D. Las teclas + y − imprimen E y F. El punto escribe una coma y ENTER genera una sentencia DATA, que tiene utilidad junto con el comando AUTO. Además las comas aparecen automáticamente cada dos caracteres, excepto cuando se usa una tecla que no sea del teclado hexadecimal.

Programa PERFECTO-128:

ESPECIAL HEX está preparada para añadirse el programa de control PERFECTO-128. Cargar este programa, quitar el NEW de la línea 16 y ejecutarlo. Teclear entonces ESPECIAL HEX y salvar el nuevo programa con otro nombre. Cuando se emplee este nuevo corrector se instalará el Especial Hexadecimal que valdrá también para los listados de C-64 puesto que los números de control generados por PERFECTO-128 son iguales a los de PERFECTO-64.

Para más comodidad se debería recortar un papel que se adaptase al perfil del teclado numérico y que indicará los nuevos caracteres de las teclas.

Al hacer STOP-RESTORE se recuperan las teclas de función normales. Se reactiva el teclado hexadecimal con SYSDEC («1515»).

```
PROGRAMA: ESPECIAL HEX

16 SYS 5120:POKE5270,18            .46
400 :                              .121
410 PRINT"[CRSRD]ESPERA...":RESTORE .125
500:A=DEC("1515"):B=DEC("1626"):S=0
:FORI=ATOB:READA$:J=DEC(A$):S=S+J:P
OKEI,J:NEXT
```

Con el entrenador del chip VDC se pueden descubrir los secretos de los gráficos del C-128.

VDC USER

Se puede mejorar la presentación de los programas propios usando letra cursiva y negrita. Esta rutina proporciona estos tipos en la pantalla de 80 columnas del C-128. Además permite cargar un juego de caracteres distinto en el VDC o modificar un patrón individual.

Tres Funciones Básicas:

Se maneja a través de tres SYS que se documentan en el mismo listado. Estas sentencias se usan igual que en el C-64 excepto por el ':=' que se pone entre la dirección y los parámetros. Por si parece complicado está el programa DEMO.VDC USER para que se vea que no tiene ninguna dificultad.

Las cajas:

En el VDC los caracteres inversos, o en negativo, se pueden obtener por medio de los bytes de atributos. Además estos bytes tienen un bit que permite visualizar un juego de caracteres alternativo. Esto significa que es posible usar las dos cajas que en el VIC sólo podemos tener por separado. Si consideramos que los caracteres inversos están de más, disponemos de 4 cajas de 128 caracteres cada una. No obstante, para no modificar los símbolos gráficos que llevan las cajas de letra mayúscula, el programa de las 4 anteriores. Estas se numeran de 0 a 7.

Utilización del programa:

Primero se mueven los caracteres de la RAM o de la ROM a las cajas. Después se pueden modificar los signos gráficos para crear nuevos caracteres que se vayan a usar en una aplicación concreta, generalmente letras de uso normal en nuestro idioma que no se incluyen en el C-128. Luego se genera por software los tipos cursiva y negrita o ambos a la vez. Ahora se pueden cambiar más gráficos inutilizados al haber sido variados o que se repiten.

Así pues, al cambiar de caja con COMMCHIFT o RVSON, apareceran negritas y cursivas. En inverso se obtiene a través del registro $F1 como indican las instrucciones.

Las rutinas se sitúan desde $1300 hasta $1443 incluyendo la zona de trabajo. Una vez ejecutadas puede ocuparse este área tranquilamente.

Programas complementarios independientes:

La obtención de los inversos puede hacerse más cómoda con el miniprograma CONTROL AC/BD. Permiten abrir los inversos dentro del modo comillas con CONTROL-A y cerrarlos con CONTROL-C, equivalentes a CHR$(1) y CHR$(3). También capacita para desactivar el subrayado con CONTROL-D. Recordar que éste se activa con CONTROL-B o CHR$(2).

El modo fast se anula cada vez que se pulsa STOP-RESTORE. Este inconveniente se soluciona con RESTORE FAST.

```
PROGRAMA: VDC.USER

1 REM COMMODORE WORLD 61              .215
2 :                                   .234
3 REM RAFAEL LOPEZ RUBIO              .53
5 :                                   .237
10 I=DEC("1300"):READA$:DOUNTILA$=" .174
FIN":D=DEC(A$):POKEI,D:I=I+1:T=T+D:
READA$:LOOP:READA
20 IFT<>ATHENPRINT"ERROR EN DATAS": .168
STOP
30 PRINT"[CLR][6CRSRD]":A$="S"+CHR$ .20
(34)+"BL.VDC USER"+CHR$(34)+",":B$=
",1300,1437:":PRINTA$"8"B$"DISCO":P
RINTA$"1"B$"CASSETTE[HOM]":MONITOR
100 DATA 20,80,03,20,80,03,20,D7,77 .92
110 DATA 20,C7,8C,AD,67,00,AE,66,00 .208
120 DATA 85,FA,86,FB,E8,E8,8E,40,14 .48
130 DATA 60,20,F1,87,BD,2F,14,85,FD .80
140 DATA 18,69,04,8D,41,14,A9,00,85 .220
150 DATA FC,60,A2,12,A5,FD,20,CC,CD .224
160 DATA E8,A5,FC,4C,CC,CD,20,00,13 .66
170 DATA 20,F1,87,8E,3F,14,20,1C,13 .254
180 DATA 20,45,A8,20,2F,13,A0,00,AE .108
190 DATA 3F,14,A9,FA,20,74,FF,20,CA .4
200 DATA CD,C8,C0,08,90,F0,A9,00,20 .118
210 DATA CA,CD,88,D0,FA,AD,40,14,D0 .122
220 DATA 01,60,18,A5,FA,69,08,85,FA .38
230 DATA 90,D9,E6,FB,A5,FB,CD,40,14 .92
240 DATA 90,D0,60,20,00,13,20,1C,13 .186
250 DATA A6,FA,CA,30,0D,A9,10,18,65 .116
260 DATA FC,85,FC,90,F4,E6,FD,B0,F0 .39
270 DATA A0,00,98,48,20,F1,87,68,A8 .229
280 DATA 8A,99,37,14,C8,C0,08,D0,F0 .27
290 DATA A0,0F,8C,3F,14,A0,00,8C,40 .179
```

```
300 DATA 14,A9,37,A2,14,85,FA,86,FB .249
310 DATA 20,45,A8,4C,4B,13,4E,37,14 .147
320 DATA 4E,38,14,4E,39,14,0E,3C,14 .101
330 DATA 0E,3D,14,0E,3E,14,60,A2,07 .175
340 DATA BD,37,14,8D,42,14,4A,0D,42 .101
350 DATA 14,9D,37,14,CA,10,F0,60,A0 .41
360 DATA 00,20,D8,CD,99,37,14,C8,C0 .39
370 DATA 08,D0,F5,60,20,80,03,20,1C .73
380 DATA 13,20,F1,87,8E,43,14,20,45 .47
390 DATA A8,20,2F,13,20,E9,13,AD,43 .123
400 DATA 14,F0,05,20,D6,13,30,03,20 .201
410 DATA C3,13,20,AB,13,18,A5,FC,69 .55
420 DATA 10,85,FC,90,E1,E6,FD,A5,FD .181
430 DATA CD,41,14,90,D8,60,20,24,28 .67
440 DATA 2C,30,34,38,3C              .225
450 DATA FIN, 33805                  .21
460 :                                .181
470 INSTRUCCIONES                    .119
480 :                                .201
490 SYSDEC("133C"):=DIRECCION,BANCO .27
,CAJA
500 TRASLADA CARACTERES A UNA CAJA  .193
DE LA VRAM
510 DIRECCION:DE LOS CARACTERES EN  .1
RAM O ROM (1024-64768)
520 BANCO:DE LOS CARACTERES (0-15)  .112
530 CAJA:DE LA VRAM (0-7)           .162
540 :                               .6
550 SYSDEC("1381"):=CARACTER,CAJA,D .168
0,D1,D2,D3,D4,D5,D6,D7
560 MODIFICA UN CARACTER INDIVIDUAL .230
 EN LA VRAM
570 CARACTER:CODIGO DE PANTALLA (0- .20
255 PARA CAJAS 0 Y 4)
580 CAJA:DE LA VRAM (0-7)           .212
590 D0-D7:DATOS DEL CARACTER        .46
600 :                               .66
610 SYSDEC("13F7"):=CAJA,TIPO       .152
620 MODIFICA EL TIPO DE LA CAJA     .168
630 CAJA:DE LA VRAM (0-7)           .6
640 TIPO:0=CURSIVA;1=NEGRITA        .142
650 :                               .116
660 NO OLVIDAR EL ':=' DESPUES DEL  .204
SYS
670 :                               .136
680 INVERSOS : POKEDEC("F1"),PEEK(D .230
EC("F1"))OR64
690 DIRECTOS : POKEDEC("F1"),PEEK(D .210
EC("F1"))AND191
```

PROGRAMA: DEMO-VDC-USER

```
1 REM COMMODORE WORLD 61             .215
2 :                                  .234
3 REM RAFAEL LOPEZ RUBIO             .53
5 :                                  .237
100 REM DEMO VDC USER                .26
110 FAST:GRAPHIC5:GOSUB330:BLOAD"BL .120
.VDC USER"
120 :                                .96
130 REM TRASLADA CARACTERES EN ROM  .44
A LAS CAJAS
140 FORI=0TO6STEP2:SYSDEC("133C"):= .198
DEC("D800"),14,I:SYSDEC("133C"):=DE
C("DA00"),14,I+1:NEXT
150 :                                .126
160 REM GENERA CARACTERES PROGRAMAD .96
OS
170 DATA 7C,00,7C,66,66,66,66,00,10 .228
1
180 DATA 18,00,18,30,60,66,3C,00,11 .112
6
190 DATA 18,00,00,18,18,18,18,00,11 .216
7
200 DATA 0E,1F,1F,1F,0F,07,03,01,97 .238
210 DATA 70,F8,F8,F8,F0,E0,C0,80,11 .194
8
220 DATA 7E,00,66,76,7E,6E,66,00,12 .246
7
230 DATA 00,00,3C,66,60,66,3C,18,10 .62
6
240 FORC=0TO6STEP2:RESTORE170:FORI= .70
0TO6:FORJ=0TO7:READK$:D(J)=DEC(K$):
NEXT:READCH:SYSDEC("1381"):=CH,C,D(
0),D(1),D(2),D(3),D(4),D(5),D(6),D(
7):NEXTI,C
250 :                                .226
260 REM GENERA TIPOS                 .7
270 DATA 2,0,3,0,4,1,5,1,6,0,7,0,6, .245
1,7,1
280 FORI=0TO7:READC,T:SYSDEC("13F7" .167
):=C,T:NEXT
290 :                                .11
300 END                              .47
310 :                                .31
320 REM IMPRIME DEMO                 .177
330 SCNCLR:M$(0)=CHR$(142):M$(1)=CH .63
R$(142)+"[RVSON]":M$(2)=CHR$(14):M$
(3)=CHR$(14)+"[RVSON]"
340 C$(0)="@ABCDEFGHIJKLMNOPQRSTUVW .95
XYZ[[LIBRA]][FLCH ARRIBA]_"
350 C$(1)=" !"+CHR$(34)+"#$%&'()*+, .17
-./0123456789:;<=>?"
360 C$(2)="[SHIFT*][SHIFTA][SHIFTB] .67
[SHIFTC][SHIFTD][SHIFTE][SHIFTF][SH
IFTG][SHIFTH][SHIFTI][SHIFTJ][SHIFT
K][SHIFTL][SHIFTM][SHIFTN][SHIFTO][
SHIFTP][SHIFTQ][SHIFTR][SHIFTS][SHI
FTT][SHIFTU][SHIFTV][SHIFTW][SHIFTX
][SHIFTY][SHIFTZ][SHIFT+][COMM-][SH
IFT-][PI][COMM*]"
370 C$(3)="[SHIFT SPC][COMMK][COMMI .221
][COMMT][COMM@][COMMG][COMM+][COMMM
][COMM\][SHIFT\][COMMN][COMMQ][COMM
D][COMMZ][COMMS][COMMP][COMMA][COMM
E][COMMR][COMMW][COMMH][COMMJ][COMM
L][COMMY][COMMU][COMMO][SHIFT@][COM
MF][COMMC][COMMX][COMMV][COMMB]"
380 COLOR6,16:FORI=0TO3:FORJ=0TO3:P .213
RINTM$(I)"[WHT][SHIFTC][SHIFTA][SHI
FTJ][SHIFTA]"INT(I*2+J/2)":[BLK] "C
$(J):NEXTJ,I
390 PRINTM$(0)"[CRSRD][BLK][SHIFTP] .175
UEDES RESALTAR TEXTOS EN [RVSON]CUR
```

PROGRAMA: CONTROL A/B

```
1 REM COMMODORE WORLD 61             .215
2 :                                  .234
3 REM RAFAEL LOPEZ RUBIO             .53
5 :                                  .237
10 FORI=3072TO3134:READA$:D=DEC(A$) .80
:S=S+D:POKEI,D:NEXT:IFS<>7318THENPR
INT"ERROR!":STOP:ELSESYS3072
20 DATA A9,0B,8D,26,03,A9,0C,8D,27, .128
03,60,48,A5,9A,C9,03,D0,29,A5,F4,D0
,25,A5,F5,D0,21,68,48,C9,01,F0,15
30 DATA C9,03,F0,0A,C9,04,D0,13,20, .162
CE,C8,18,90,0D,A5,F1,29,BF,18,90,04
,A5,F1,09,40,85,F1,68,4C,79,EF
```

PROGRAMA: RESTORE.FAST

```
1 REM COMMODORE WORLD 61             .215
2 :                                  .234
3 REM RAFAEL LOPEZ RUBIO             .53
5 :                                  .237
100 DATA A2,F2,A0,07,8E,00,0A,8C,01 .236
,0A,20,B6,77,4C,03,40:FORI=2024TO20
39:READA$:POKEI,DEC(A$):NEXT:BANK15
:SYS2024:REM ANULAR CON POKE2035,19
9
```

PROGRAMA: TABLERO COLOR

```
1 REM COMMODORE WORLD 61             .215
2 :                                  .234
3 REM RAFAEL LOPEZ RUBIO             .53
5 :                                  .237
10 REM *** INICIALIZACION VARIABLES .128
 (GENERAL:I,I$,J,J$,K,K$,L,N,O)
20 REM * PANTALLA (C-64)/COLOR BORD .158
E,FONDO,TINTA,CAJA ALTA
30 POKE53280,13:POKE53281,15:PRINT" .220
[BLK][CLR]"CHR$(142)CHR$(8)
40 REM * GRAFICOS                    .146
50 I$="[CRSRD][2CRSRL]":F$(0)="[RVS .4
ON][COMMC][COMMV][RVSOFF]"+I$+"[COM
M*][SHIFT\][BLK]":F$(1)="[RVSON][SH
IFT\][COMM*][RVSOFF]"+I$+"[COMM*][S
HIFT\][BLK]":F$(2)="[RVSON][SHIFT\]
[COMM*]"+I$+"[COMMF][COMMD][RVSOFF]
[BLK]"
60 F$(3)="[RVSON][COMMV][COMMC][RVS .184
```

Con «Especial HEX» se pueden introducir más comodamente los listados hexadecimales.

VDC

User permite la utilización de letras en cursiva y en negrita en la pantalla de 80 columnas.

```
OFF]"+I$+"[SHIFT@][SHIFTL][BLK]":FO
RI=1TO5:C$(I-1)=MID$("[WHT][YEL][RE
D][BLU]"+CHR$(0),I,1):NEXT:V$="[COM
M6]"
70 F$(4)=CHR$(0)+"[SHIFTO][SHIFTP]"  .124
+I$+"[SHIFTL][SHIFT@][BLK]":SP$="[1
0SPC]":C$=CHR$(13):F$="":CL$="":DI$
=""
80 REM * CURSOR                      .240
90 A$="[HOM]":FORI=0TO24:A$=A$+"[CR  .196
SRD]":NEXT:X=0:Y=0:MX=18:MY=13:XX=0
:YY=0:XI=0:YI=0:M$="
100 REM * DEL JUEGO                  .146
110 DIMM(63),T(7,7):M=64:A=240:B=15  .184
:I=0:FORJ=0TO3:FORK=0TO3:FORL=0TO3
120 M(I)=K*16+L:I=I+1:E(J)=J:NEXTL,  .84
K,J:FORI=0TO7:FORJ=0TO7:T(I,J)=68:N
EXTJ,I
130 REM * MENU,MENSAJES              .0
140 NI=1:ME=1:T$=" <TECLA>[2SPC]":Q  .86
$(1)=" <Q>=SALIR":Q$(0)="[2SPC]-TIE
MPO-"
150 FORI=0TO4:M$(I)="NIVEL"+STR$(I)  .238
:NEXT:M$(5)=" AYUDA "
160 K$=". TIEMPO LIMITADO PARA PONE  .182
R UNA FIGURA":J$=". PULSA <J> PARA
JUGAR."
170 I$(0)="NIVEL DE JUEGO EN EL QUE  .32
 SE PUEDE DEVOLVER UNA FIGURA AL MA
ZO"+J$
180 I$="COLOCA BIEN LAS FIGURAS SEG  .122
UN VAN SALIENDO":I$(1)=I$+J$:I$(3)=
I$+K$+J$
190 I$="APARECEN 4 FIGURAS Y COLORE  .88
S. COLOCA FIGURAS IGUALES SOBRE LAS
 FIGURAS"
200 I$=I$+" Y COLORES IGUALES SOBRE  .54
 LOS COLORES":I$(2)=I$+J$:I$(4)=I$+
K$+J$
210 I$="DEBES PONER 64 FIGURAS SOBR  .36
E EL TABLERO A MEDIDA QUE SALEN DE
MODO "
220 I$=I$+"QUE ALREDEDOR DE CUALQUI  .104
ER FIGURA, EN LOS LADOS O LAS DIAGO
NALES, "
230 I$(5)=I$+"NO HAYA OTRA FIGURA C  .104
ON SU MISMO COLOR O SU MISMO DIBUJO
."
240 REM * TECLADO,AZAR Y TIEMPOS     .86
250 GE$="":G$="":G=0:DEFFNF(I)=INT(  .76
RND(1)*I):S(3)=450:S(4)=510:S=0
260 REM *** PROGRAMA PRINCIPAL       .167
270 GOSUB590:REM * MENU E INSTRUCCI  .33
ONES:
280 X=29:Y=9:GOSUB570:PRINT"[RVSON]  .23
"SP$:FORME=0TO5:GOSUB670:NEXT:PRINT
SPC(X)"[RVSON]"SP$
290 X=29:GE$="[CRSRU]"+C$+"[CRSRD]"  .89
:GOSUB630:GE$="J":ME=NI:J=G-2:IFG=0
THEN290
300 IFJTHENNI=NI+J-(J>0)*(NI=5)+(J<  .27
0)*(NI=0):GOSUB670:ME=NI:GOSUB670:G
OTO290
310 X=0:Y=2:GOSUB570:FORI=1TOLEN(I$  .215
(NI))STEP10:PRINT" [RVSON]"MID$(I$(
NI),I,10):NEXT
320 GOSUB630:IFG=0ORNI=5THENX=1:FOR  .47
Y=2TO23:GOSUB570:PRINTSP$:NEXT:GOTO
290
330 REM * NIVELES DE JUEGO           .227
340 GOSUB590:GOSUB710:X=1:Y=2:GOSUB  .137
570:PRINTM$(NI)". USA <CRSRS><*><CR
>.";
350 PRINTQ$(-(NI<3)):IFNI<2ORNI=3TH  .149
EN420
360 REM * 4 FIGURAS-4 COLORES NIVEL  .135
ES 2,4
370 FORI=4TO1STEP-1:J=FNF(I):F=E(J)  .17
:E(J)=E(I-1):X=12-(I>2)*14:Y=7+(IAN
D1)*14
380 GOSUB690:PRINT"[BLK]"DI$:T((X-1  .93
2)/2,(Y-7)/2)=F+64:NEXT:FORI=0TO3:E
(I)=I:NEXT
390 FORI=4TO1STEP-1:J=FNF(I):F=E(J)  .9
:E(J)=E(I-1):X=14-(I>2)*10:Y=9+(IAN
D1)*10
400 F=F*16:GOSUB690:PRINTCL$F$(4):T  .91
((X-12)/2,(Y-7)/2)=F+4:NEXT
410 REM * JUEGO                      .191
420 I=FNF(M):F=M(I):M=M-1:M(I)=M(M)  .235
:Y=4:X=19:GOSUB690:PRINTF$:S=TI:L=F
430 GE$=C$+"Q*[CRSRR][CRSRL][CRSRD]  .187
[CRSRU]":GOSUB620:ONG+1GOTO430,450,
440,440:G=G-3:GOSUB720:GOTO430
440 ONG-1GOSUB780,710:GOTO430        .225
450 J=T(XX,YY):IFNI=0ANDJ<>68THENM(  .109
M)=J:M=M+1:T(XX,YY)=68:GOSUB740:GOT
O430
460 REM * COMPRUEBA                  .47
470 ON2+((BANDJ)<>4AND(AANDJ)<>64)G  .1
OTO430:ON2+(J=68)GOTO490:I=AANDJ:K=
BANDJ
480 ON2+(I=64AND(BANDJ)<>(BANDL))GO  .107
TO790:IFK=4AND(AANDJ)<>(AANDL)THEN8
00
490 FORI=XX>0TO-(XX<7):FORJ=YY>0TO-  .113
(YY<7):K=T(XX+I,YY+J)OR-255*(I*J=I+
J)
500 ON2+((BANDK)=(BANDL))GOTO790:ON  .245
2+((AANDK)=(AANDL))GOTO800:NEXTJ,I
510 T(XX,YY)=L:GOSUB740:ON2+(NI>2)G  .1
OSUB710:ON1+(M=0)GOTO420:IFNI<>4THE
N770
520 REM * CONSEGUIDO NIVEL 4         .132
530 X=0:Y=2:GOSUB570:FORI=0TO4:PRIN  .88
TSP$SP$SP$"[9SPC]":NEXT:X=12:Y=4
540 GOSUB570:PRINT"JUEGO CONSEGUIDO  .24
":PRINTSPC(12)"[COMM6][16COMMU][BLK
]":GOTO770
550 REM *** SUBRUTINAS : ->()=ENTRA  .84
DA; ()->=SALIDA; *()=MODIFICA.
560 REM * POSICION ->(X,Y)           .2
570 PRINTLEFT$(A$,Y+1)SPC(X);:RETUR  .38
N
580 REM * TABLERO *(I,X,Y)           .218
590 PRINT"[CLR]"SPC(12)"TABLERO DE   .202
COLOR"C$" ";:FORI=0TO37:PRINT"[COMM
6][COMMU][BLK]";:NEXT
600 FORY=7TO21STEP2:FORX=12TO26STEP  .120
2:GOSUB570:PRINTF$(4):NEXTX,Y:RETUR
N
610 REM * ESPERA TECLA ->(GE$,S,NI)  .76
 (G,G$)-> *(I)
620 IFNI>2THEN640                    .150
630 I=RND(-TI):GETG$:ON2+(G$="")GOT  .142
O630:GOTO650
640 I=RND(-TI):I=TI:I=I-S:GETG$:ON2  .238
+(G$=""ANDI<S(NI))GOTO640:IFI>=S(NI
)THEN810
650 G=0:FORI=1TOLEN(GE$):G=G-I*(G$=  .222
MID$(GE$,I,1)):NEXT:RETURN
660 REM * 1 LINEA MENU ->(ME) *(Y)   .20
670 Y=ME+10:GOSUB570:PRINT"[RVSON]   .150
[RVSOFF]"CHR$(18-128*(ME=NI))M$(ME)
"[RVSON][2SPC][RVSOFF]":RETURN
680 REM * FIGURA ->(F,X,Y) (F$,CL$,  .222
DI$,POS.)->
690 CL$=C$(F/16):DI$=F$(BANDF):F$=C  .156
L$+DI$:GOSUB570:RETURN
700 REM * CURSOR ->(G) (XX,YY,CUR.)  .82
-> *(X,Y,F)
710 XX=2:YY=3:XI=0:G=1:REM * CENTRO  .2
720 IFG<3THENYI=0:XI=1+2*(G=2)-(XX=  .172
7)*(G=1)+(XX=0)*(G=2):GOTO740:REM *
 HORIZ.
730 XI=0:YI=1+2*(G=4)-(YY=7)*(G=3)+  .232
(YY=0)*(G=4):REM * VERT.
740 X=MX:Y=MY:GOSUB570:PRINTM$:XX=X  .224
X+XI:YY=YY+YI:X=12+XX*2:Y=7+YY*2
750 XI=0:YI=0:MX=X:MY=Y:F=T(XX,YY):  .4
GOSUB690:M$=F$:PRINTV$DI$:RETURN
760 REM *** FIN NIVEL ->(NI) *(X,Y,  .96
I$,J$)
770 I$=" MUY BIEN ":J$=T$:V$=CL$:ON  .103
2+(NI=4)GOSUB740:GOTO820
780 I$=" SHIFT<F> ":J$=" PARA FIN "  .191
:ON2+(NI<3)GOTO820:RETURN
790 I$="FIGURA MAL":J$=T$:GOTO820    .115
800 I$="COLOR[2SPC]MAL":J$=T$:GOTO8  .175
20
810 I$="[2SPC]TIEMPO[2SPC]":J$=T$:G  .103
OSUB600
820 X=29:Y=21:GOSUB570:PRINT"[RVSON  .171
]"I$C$SPC(X)"[RVSON]"J$:GOSUB630:IF
J$=T$ORG$="[SHIFTF]"THENRUN
830 GOSUB570:PRINTSP$C$SPC(X)SP$:RE  .145
TURN
```

Por Michael Broussard

KALA

¿Tienes las ideas suficientemente claras como para ganar al ordenador en este milenario juego de estrategia?

Kalah, escrito completamente en Basic para el C-64 o el C-128 (en modo 40 columnas) es un juego de estrategia basado en un popular pasatiempo africano en el que se juega con cuentas, normalmente guisantes, sobre 14 hoyos excavados en la tierra. Para poder jugar, teclea en primer lugar el programa del listado 1, y grábalo a disco antes de ejecutarlo. El ordenador es tu oponente.

Cuando tecleas RUN, Kalah borra la pantalla y te pregunta el nivel en el que quieres jugar: Principiante, Intermedio o Experto. Para elegir, teclea 1, 2 ó 3 respectivamente y pulsa la tecla RETURN. La primera vez es conveniente que pruebes con 1, Principiante. Más adelante trataré algunas de las variadas estrategias de Kalah de acuerdo con el nivel de juego elegido.

Preparativos

A continuación, Kalah te pide que introduzcas el número de cuentas (llamadas «piedras») que quieres colocar en cada hoyo al comenzar. Si pulsas return se elige el número por defecto, que es tres. (Se puede seleccionar cualquier número entre 1 y 6, como se explicará más adelante). Tras realizar la selección, la pantalla se borra y aparece el «tablero» del juego, que consiste en 14 cuadrados, representando los hoyos del juego original (ver figura 1).

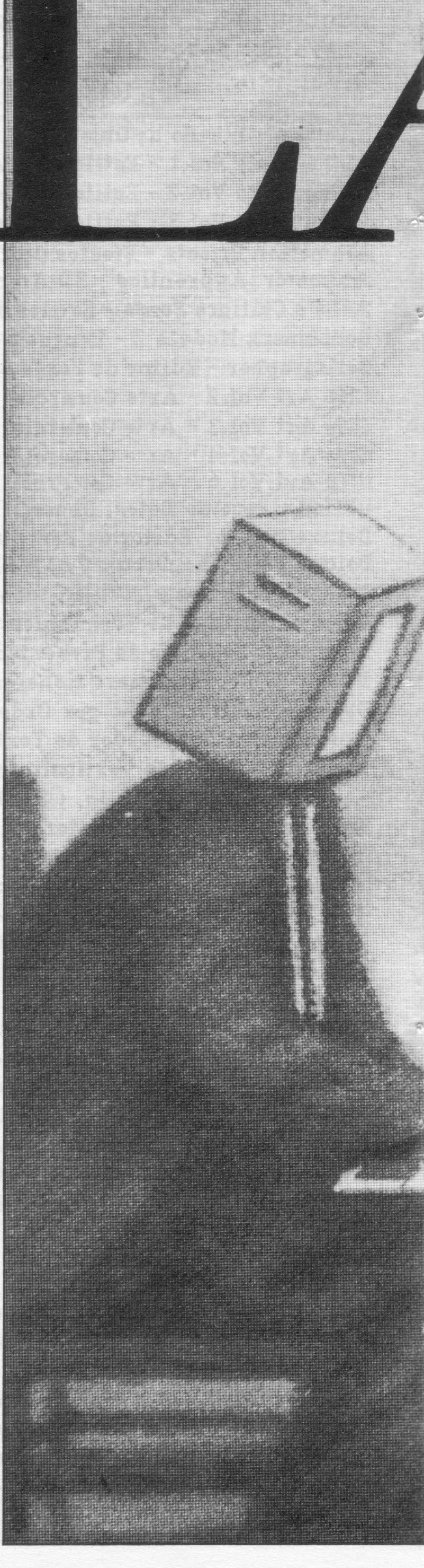
En cada hoyo hay un número que indica cuántas piedras contiene. Si no aparece ningún valor, ese hoyo está «vacío». Los seis hoyos de la parte inferior de la pantalla, marcados con las letras A-F están bajo tu control. Los seis de arriba, G-L, están controlados por el ordenador. El hoyo que hay ente las dos filas en la parte derecha es tu «hoyo-casa», y el de la izquierda el del ordenador. El objetivo del juego es acumular más piedras en tu «casa» que las que consiga el ordenador.

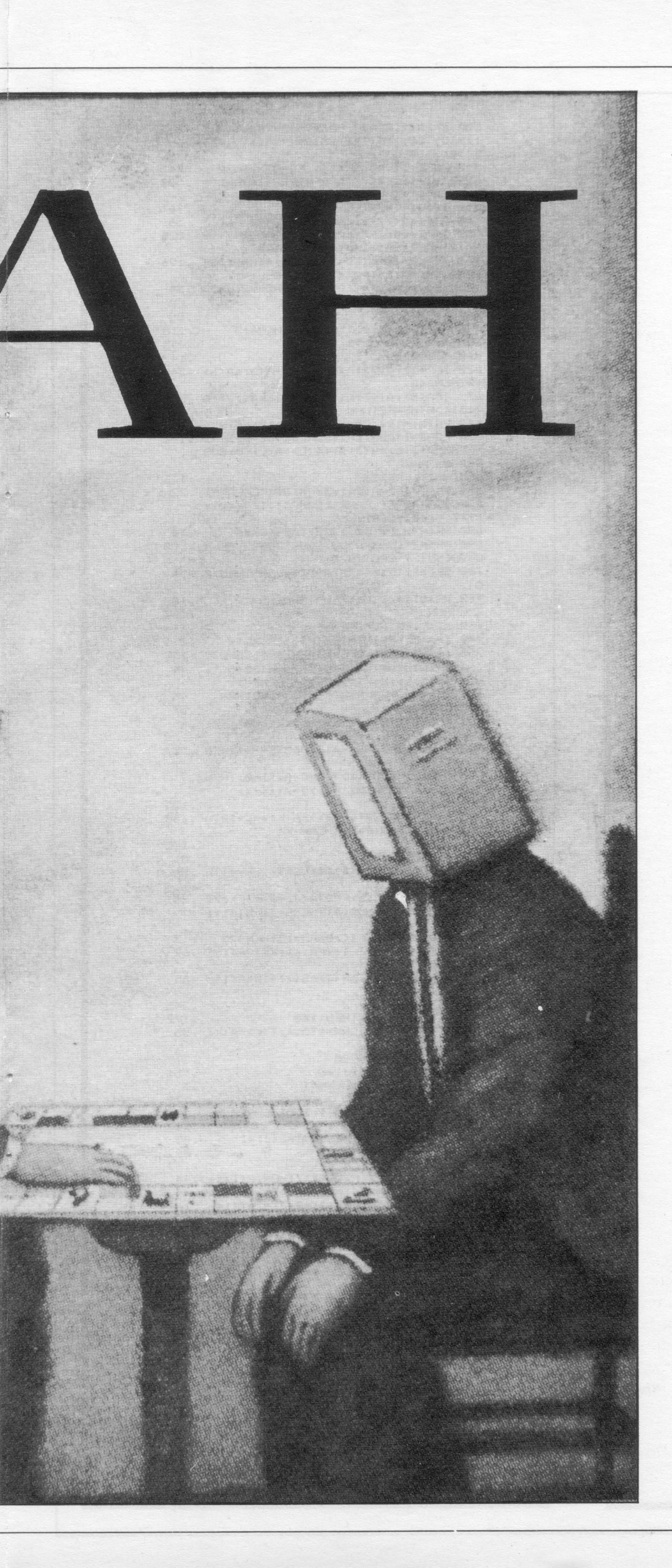
Cómo jugar

El ordenador decide aleatoriamente quién juega primero. En el juego real, moverías tomando todas las piedras de uno de los hoyos y redistribuyéndolas en los hoyos adyacentes, siempre en el sentido contrario al de las agujas del reloj, sin saltarte ningún hoyo excepto el hoyo-casa de tu oponente. En el juego de ordenador, el principio es el mismo, sólo que es la máquina la que hace los movimientos por ti.

Para seleccionar desde qué hoyo quieres jugar, simplemente teclea la letra correspondiente (de la A a la F) y pulsa la tecla RETURN. Utiliza la tecla DELETE si te equivocas y quieres rectificar. Cuando haces una se-

El objetivo del juego es acumular más piedras en tu «casa» que las que consiga el ordenador.





En el juego de ordenador, el principio es el mismo, sólo que es la máquina la que hace los movimiento por ti.

lección y pulsas RETURN, los números de los cuadrados de la pantalla cambian para reflejar la nueva situación. Los movimientos del ordenador son similares, desde los hoyos G a L, sólo que él salta tu hoyo-casa.

Por ejemplo, si tu primer movimiento es el A, las tres piedras de hoyo A desaparecen y son depositadas en los hoyos B, C y D, que quedan cada uno con cuatro piedras. Aquí termina tu turno. Si tu primer movimiento fuera desde el hoyo E, las piedras quedarían en F, tu hoyo-casa y el hoyo G del ordenador. Otro ejemplo, si jugaras E y en E hubiera diez piedras, éstas irían a F, tu hoyo-casa, de la G a la L, A y B.

Observa cómo puedes dejar piedras en tu propia casa cuando pasas por allí, pero que tienes que saltarte el hoyo de tu oponente. Además, las piedras nunca pueden jugarse desde las casas. Una vez que llegan allí permanecen hasta el final del juego.

Reglas adicionales

Dos reglas adicionales forman gran parte de la estrategia del juego. La primera es que si haces una jugada en la que la última piedra quede depositada en tu propia casa, obtienes inmediatamente otro turno. Jugando con cuidado, puedes conseguir varios turnos seguidos.

La segunda regla es que si la última piedra que juegas termina en un hoyo vacío de los que te pertenecen, todas las piedras del hoyo del adversario que se encuentren en el lugar de enfrentarse son «capturadas» y pasan directamente a tu casa, aumentando así tu puntuación. Por ejemplo, supón que tu hoyo C está vacío y que A contiene dos piedras. Al jugar desde A, la primera piedra queda en B y la segunda en C, un hoyo que estaba vacío. En este caso, capturarías todas las piedras de tu oponente que se encontraran en J, que es el hoyo opuesto al C. Todas las piedras de J (si las hubiera) pasan a tu propia casa, y termina tu turno.

La jugada continúa hasta que uno de los jugadores no tiene más piedras en su lado del tablero. El juego termina, y todas las piedras que quedaran en el lado

KALAH

```
PROGRAMA: KALAH                  LISTADO 1

1 REM KALAH                             .171
2 REM (C)1989 BY M.BROUSSARD            .232
3 REM (C)1989 BY COMMODORE WORLD        .19
4 :                                     .236
100 T1=0:REM RETARDO ENTRE TURNOS       .48
110 DEF FNB(X)=X+14*(X>14)-1*(X>20)     .10
+14*(X>27)-1*(X>33)+14*(X>40)-1*(X>
46)
120 POKE53280,3:POKE53281,1:PRINT"[     .84
BLK][CLR][CRSRD]"TAB(18)"[RVSON]KAL
AH[2CRSRD]"
130 R$="[HOM]":FORK=1TO80:C$=C$+"[C     .176
RSRR]":R$=R$+"[CRSRD]":NEXT
140 DIMR(14),C(14),A(14),S(14),T(14     .210
):FORK=1TO14:READR(K),C(K):NEXT
150 PRINT"ELIGE NIVEL DE JUEGO:[CRS     .234
RD]":PRINTTAB(5)"1) PRINCIPIANTE"
160 PRINTTAB(5)"2) INTERMEDIO":PRIN     .54
TTAB(5)"3) EXPERTO"
170 PRINT"[CRSRD]"TAB(5):INPUT"(1,      .144
2, O 3)[2SPC]2[3CRSRL]";N$:LV=VAL(N
$):IFLV<1ORLV>3THEN150
180 N$="":INPUT"[CRSRD]CUANTAS PIED     .220
RAS POR HOYO[2SPC]3[3CRSRL]";N$:N=V
AL(N$)
190 IFN<1ORN>6THENPRINT"ELIGE UN VA     .160
LOR ENTRE 1 Y 6!":GOTO180
200 X$="[5SPC]":PRINT"[CLR][CRSRD]"     .186
TAB(18)"[RVSON]KALAH[2CRSRD]":F=1
210 PRINTX$"[2SPC]L[4SPC]K[4SPC]J[4     .32
SPC]I[4SPC]H[4SPC]G"
220 PRINTX$"[SHIFTU][3SHIFT*][SHIFT     .190
I][SHIFTU][3SHIFT*][SHIFTI][SHIFTU]
[3SHIFT*][SHIFTI][SHIFTU][3SHIFT*][
SHIFTI][SHIFTU][3SHIFT*][SHIFTI][SH
IFTU][3SHIFT*][SHIFTI]
230 PRINTX$"[SHIFT-][3SPC][2SHIFT-]     .192
[3SPC][2SHIFT-][3SPC][2SHIFT-][3SPC
][2SHIFT-][3SPC][2SHIFT-][3SPC][SHI
FT-]
240 PRINTX$"[SHIFTJ][3SHIFT*][SHIFT     .88
K][SHIFTJ][3SHIFT*][SHIFTK][SHIFTJ]
[3SHIFT*][SHIFTK][SHIFTJ][3SHIFT*][
SHIFTK][SHIFTJ][3SHIFT*][SHIFTK][SH
IFTJ][3SHIFT*][SHIFTK]
250 PRINT"[2SPC][SHIFTU][3SHIFT*][S     .90
HIFTI][26SPC][SHIFTU][3SHIFT*][SHIF
TI]
260 PRINT"[2SPC][SHIFT-][3SPC][SHIF     .151
T-][26SPC][SHIFT-][3SPC][SHIFT-]
270 PRINT"[2SPC][SHIFTJ][3SHIFT*][S     .87
HIFTK][26SPC][SHIFTJ][3SHIFT*][SHIF
TK]
280 PRINTX$"[SHIFTU][3SHIFT*][SHIFT     .251
I][SHIFTU][3SHIFT*][SHIFTI][SHIFTU]
[3SHIFT*][SHIFTI][SHIFTU][3SHIFT*][
SHIFTI][SHIFTU][3SHIFT*][SHIFTI][SH
IFTU][3SHIFT*][SHIFTI]
290 PRINTX$"[SHIFT-][3SPC][2SHIFT-]     .253
[3SPC][2SHIFT-][3SPC][2SHIFT-][3SPC
][2SHIFT-][3SPC][2SHIFT-][3SPC][SHI
FT-]
300 PRINTX$"[SHIFTJ][3SHIFT*][SHIFT     .149
K][SHIFTJ][3SHIFT*][SHIFTK][SHIFTJ]
[3SHIFT*][SHIFTK][SHIFTJ][3SHIFT*][
SHIFTK][SHIFTJ][3SHIFT*][SHIFTK][SH
IFTJ][3SHIFT*][SHIFTK]
310 PRINTX$"[2SPC]A[4SPC]B[4SPC]C[4     .137
SPC]D[4SPC]E[4SPC]F"
320 FORK=1TO14:A(K)=N:NEXT:A(7)=0:A     .19
(14)=0:K=1:GOSUB730
330 X$="TU":F=1:IFRND(0)<.5THENX$="     .165
YO":F=2
340 PRINTMID$(R$,1,19)TAB(12)"PRIME     .31
R MOVIMIENTO: "X$:FORK=1TO1600:NEXT
K
350 PRINTMID$(R$,1,19)"[39SPC]";:IF     .187
F=2THEN440
360 PRINTMID$(R$,1,19)TAB(14)"TU JU     .35
GADA?[6SPC][5CRSRL]";:X$=""
370 GETZ$:IFZ$=""THEN370                .79
380 IFZ$=CHR$(20)ANDX$<>""THENPRINT     .101
"[CRSRL] [CRSRL]";:X$="":GOTO370
390 IFZ$=CHR$(13)ANDX$<>""THEN420       .125
400 IFZ$>="A"ANDZ$<="F"ANDX$=""THEN     .211
X$=Z$:PRINTX$;:GOTO370
410 GOTO370                             .213
420 K=ASC(X$)-64:GOSUB470               .47
430 ONFGOTO360,440                      .177
440 PRINTMID$(R$,1,22)TAB(14)"[RVSO     .31
N]PENSANDO...[RVSOFF]";:GOSUB790:K=
P
450 PRINTMID$(R$,1,22)"[39SPC]";        .51
460 PRINTMID$(R$,1,19)TAB(14)"MI JU     .169
GADA ES "CHR$(K+63):GOSUB470:X=1:GO
TO430
470 J=A(K):IFJ=0THENONFGOTO360,1160     .115
480 A(K)=0:I=K                          .183
490 X$=STR$(A(I)):IFA(I)=0THENX$="[     .197
2SPC]"
500 PRINT"[RED]":PRINTMID$(R$,1,R(I     .231
))MID$(C$,1,C(I))X$MID$(" ",1,-(A(I
)<10))"[BLU]";
510 FORZ=1TO40:NEXTZ:IFJ=0THEN550       .241
520 I=I+1:IF(I=14ANDF=1)OR(I=7ANDF=     .188
2)THENI=I+1
530 IFI>14THENI=1                       .36
540 A(I)=A(I)+1:J=J-1:GOTO490           .36
550 X=-I+14:IFF=1ANDA(I)=1ANDI<7THE     .226
NA(7)=A(7)+A(X):A(X)=0:GOTO570
560 X=-(I-14):IFF=2ANDA(I)=1ANDI>8T     .126
HENA(14)=A(14)+A(X):A(X)=0
570 IF(I=7ANDF=1)OR(I=14ANDF=2)THEN     .54
590
580 F=2+(F=2)                           .88
590 GETX$:IFX$<>""ORX>T1THEN610         .218
600 X=X+1:GOTO590                       .242
610 GOSUB730                            .142
620 IFA(1)+A(2)+A(3)+A(4)+A(5)+A(6)     .248
=0THEN640
630 IFA(8)+A(9)+A(10)+A(11)+A(12)+A     .54
(13)<>0THENRETURN
640 PRINTMID$(R$,1,21)TAB(18)"FIN"      .110
650 FORJ=1TO8STEP7:FORL=JTOJ+5:IFA(     .190
L)THENA(J+6)=A(J+6)+A(L):A(L)=0:GOS
UB730
660 NEXTL:NEXTJ                         .136
670 X=A(7):Y=A(14):X$=MID$("EMPATE[     .22
2SPC]GANASTE! GANE![2SPC]",1+-((X>Y
)+(X<Y)*2)*8,8)
680 PRINTMID$(R$,1,22)TAB(16)X$         .14
690 PRINT"[CRSRD]QUIERES JUGAR DE N     .100
UEVO[SHIFT SPC](S/N)? ";
700 GETX$:IFX$<>"S"ANDX$<>"N"THEN70     .54
0
710 PRINTX$;:IFX$="N"THENEND            .188
720 RUN                                 .252
730 K2=K:PRINT"[BLU]"                   .116
740 X=A(K):PRINTMID$(R$,1,R(K));        .216
750 X$=STR$(X):IFX=0THENX$="[2SPC]"     .158
760 PRINTMID$(C$,1,C(K))X$MID$(" ",     .46
1,-(X<10));:K=K+1:IFK>14THENK=1
770 IFK=K2THENPRINT"[BLK]":RETURN       .43
780 GOTO740                             .57
790 FORK2=1TO14:T(K2)=A(K2):NEXTK2:     .85
P=0
800 X1=0:X2=0:FORK=13TO8STEP-1:X=FN     .223
B(A(K)+K)
810 IFA(K)=13OR(X>7)AND(A(X)=0)THEN     .9
IFA(14-X)>X2THENX1=K:X2=A(14-X)
820 NEXTK                               .155
830 FORK=13TO8STEP-1:IFA(K)=-K+14TH     .163
ENIF(K>X1)AND(P=0)THENP=K
840 NEXTK                               .175
850 IFPTHENRETURN                       .81
860 FORJ=1TO14:S(J)=0:NEXT:IFLV=1TH     .59
EN1100
870 FORJ=8TO13:X=FNB(A(J)+J):IFX>7T     .231
HENIF(A(X)=0)ANDA(J)THENS(J)=S(J)+1
0*A(14-X)
880 X=A(14-J):IFX<>0ORJ=13THEN920       .13
890 FORK=1TO14-J-1:IFA(K)+K=14-JTHE     .67
NS(J)=S(J)+10*A(J)
900 IFA(K)=13THENS(J)=S(J)+10*A(14-     .63
K)
910 NEXTK                               .245
920 NEXTJ:IFLV=2THEN1100                .177
930 FORJ=8TO13:FORK=1TO14:T(K)=A(K)     .55
:NEXTK
940 IFT(J)=0THEN990                     .33
950 M=T(J):N=J:T(J)=0                   .151
960 N=N+1:IFN>14THENN=1                 .7
970 IFN=7THENN=8                        .179
980 T(N)=T(N)+1:M=M-1:IFMTHEN960        .205
990 FORK=13TO8STEP-1                    .117
1000 IFT(K)=14-KTHENS(J)=S(J)+1         .45
1010 NEXTK                              .89
1020 FORK=8TO13                         .137
1030 X=FNB(T(K)+K):IFX>7THENIF(T(X)     .44
=0)ANDT(K)THENS(J)=S(J)+2*T(14-X)
1040 IFT(K)=13THENS(J)=S(J)+2*T(14-     .164
K)
1050 X=T(14-K):IFX<>0THEN1090           .104
1060 FORL=1TO14-K-1:IFT(L)>0ANDT(L)     .246
+L=14-KTHENS(J)=S(J)-10*T(K)
1070 IFA(L)=13THENS(L)=S(L)+10*A(14     .114
-L)
1080 NEXTL                              .168
1090 NEXTK:NEXTJ                        .48
1100 X=-999:FORK=8TO13:IFS(K)>=XTHE     .198
NIFS(K)=XANDRND(-1)>.5THEN1130
1110 IFS(K)<XTHEN1130                   .98
1120 X=S(K):P=K                         .252
1130 NEXTK:IFX=-999THENRETURN           .112
1140 IFA(P)=0THENS(P)=-999:GOTO1100     .82
1150 RETURN                             .188
1160 PRINT"ERROR LOGICO!!":END          .248
1170 DATA 13,6,13,11,13,16,13,21,13     .238
,26,13,31,10,34,7,31,7,26,7,21,7,16
,7,11
1180 DATA 7,6,10,3                      .232
```

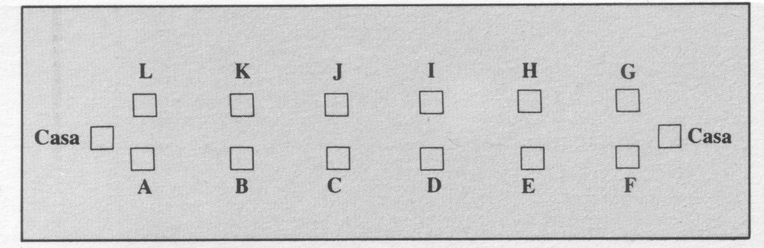


Figura 1. Cómo aparece el «tablero» del juego Kalah en la pantalla.

del oponente pasan a su casa, aumentando de este modo su puntuación (por esta razón, no siempre es conveniente «salirse» demasiado pronto). El que tuviera más piedras en su casa es el ganador del juego.

Facilidades adicionales

Mientras el ordenador está ponderando su siguiente movimiento, en la pantalla aparece el letrero «Pensado...», de modo que sabes que está haciendo algo. A medida que los jugadores efectúan sus movimientos, los hoyos afectados se vuelven momentáneamente rojos para que puedas apreciar mejor los efectos de la jugada.

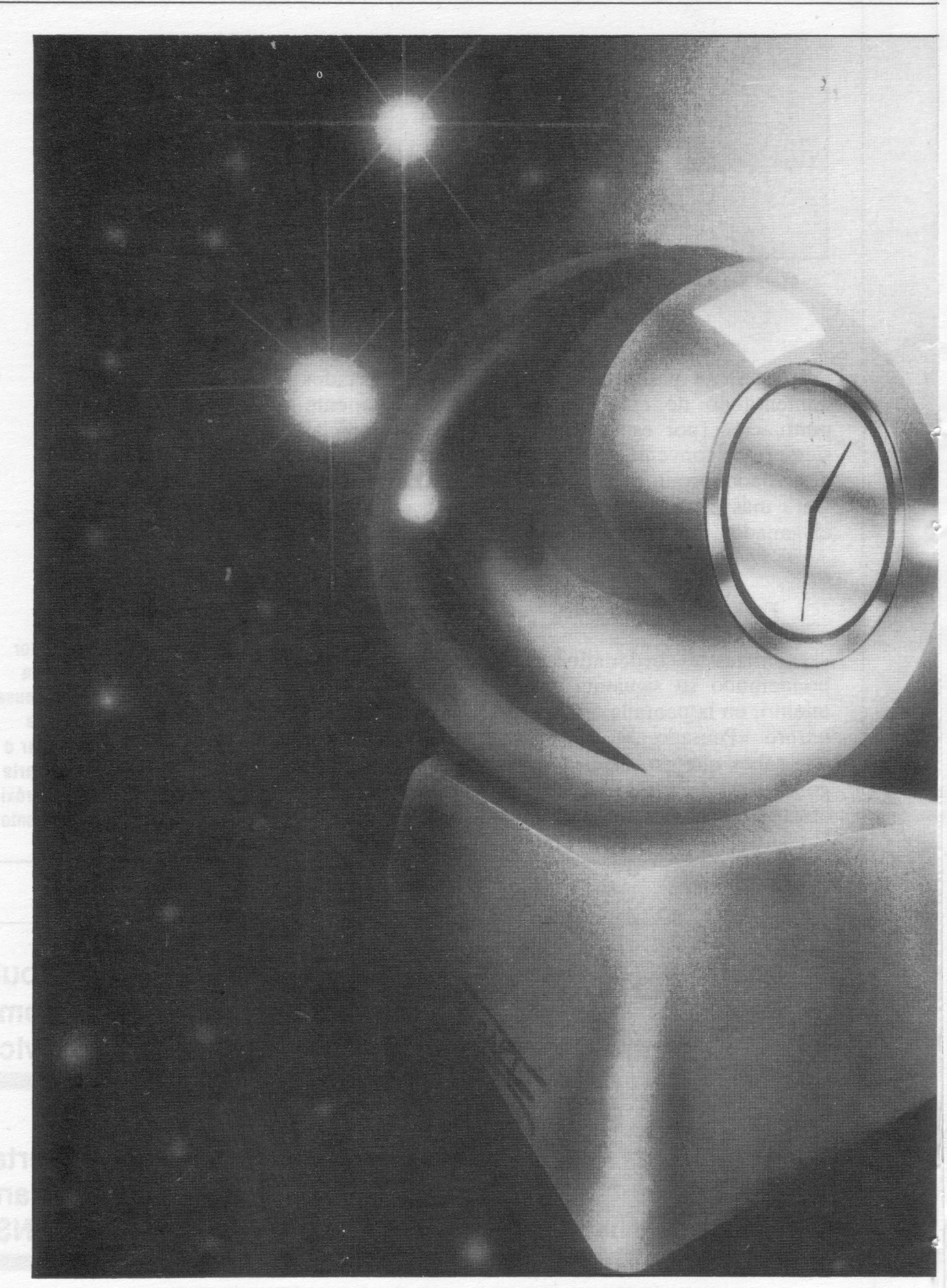
El ordenador hace una breve pausa antes de continuar o preguntarte por tu próximo movimiento. Si prefieres no esperar, puedes pulsar la barra de espacios. Alternativamente, también puedes ajustar el tiempo de la pausa modificando el valor que se asigna a la variable T1 en la línea 100 del programa. Disminuyendo este valor se acorta la pausa entre movimientos.

Una nota sobre los niveles de juego que se seleccionan al comenzar a jugar: En el nivel Principiante, el ordenador observa todos los movimientos posibles y elige cualquiera de ellos al azar. En el nivel Intermedio aumenta la estrategia, y se observan también las capturas y el evitar las capturas. Finalmente, en el modo experto, el ordenador trata de determinar qué jugada le dejará en una mejor posición con respecto a las siguientes jugadas y capturas en el siguiente movimiento.

El ordenador hace una breve pausa antes de continuar o preguntarte por tu próximo movimiento.

Tú eliges también al comienzo del juego el número de piedras por hoyo, lo que a la larga determina la calidad del juego. El valor por defecto son 3, aunque puedes seleccionar cualquier número entre 1 y 6. Al aumentar el número también aumenta la estrategia y complejidad del juego.

Las reglas de Kalah son fáciles de aprender, y pronto descubrirás que cuanto más juegas, más estrategias aprendes. Prueba y descubre por ti mismo por qué la popularidad de este juego ha perdurado a lo largo de miles de años. ■



Por Joan A. Ubach

CRONOMETRO CODIGO MAQUINA

SEGUNDA PARTE

Soy un programador aficionado del 6510, que después de leer y quedarme impresionado por el artículo aparecido en el número 38 de esta revista titulado CRONOMETRO CODIGO MAQUINA, hábilmente realizado por el Sr. Ramón Somoza, me encontré con ligeras dificultades tales como calcular el hi-bit y lo-bit de la dirección de origen de la subrutina a cronometrar, amén de la mecánica a efectuar, entrando la anterior dirección, con los pokes correctos, inicializar la rutina, y al final coger una calculadora para después de algunos peeks operarlos con una serie de sumas y multiplicaciones más o menos complejas, para obtener un resultado manufacturado. En cada una de las pruebas sobre una misma subrutina me salía diferente, debido a algún fallo cometido en esta mecánica.

Como que el programa es útil y original pero poco práctico, empecé a trabajar para adecuarlo a mis largos programas en código máquina y aquí está mi propuesta.

La rutina que presento consiste básicamente en la rutina original, a la cual he acoplado una serie de instrucciones para que faciliten su uso. Mientras en la anterior rutina la entrada de datos era totalmente manual y como tal supeditada a todo tipo de fallo humano, aquí la presento automáticamente, al arrancar el programa; el formato SYS (dirección origen), (origen subrutina a cronometrar) que es lo único que hemos de teclear con el consiguiente ahorro de tiempo y errores. Esto es así, pues llamo a las subrutinas del sistema la que comprueba como BUSCOM ($ AEDF), y evalúa expresión dejando en el FAC el resultado de esta evaluación, en este caso la dirección de comienzo de la su-



ETRO

QUINA

P A R T E

La rutina que presento consiste básicamente en la rutina original, a la cual he acoplado una serie de instrucciones para que faciliten su uso.

```
     JSR   $  AEDF
     JSR   $  AD9E
     JSR   $  B7F7
     STY   NOTA + 1
     STA   NOTA + 2
     LDA # $  FF
     STA   $  DD04
     STA   $  DD05
     STA   $  DD06
     STA   $  DD07
     LDA # $  51
     SEI
     STA   $  DD0F
     STA   $  DD0E
NOTA JSR   $  0000
     LDA # $  00
     STA   $  DD0E
     STA   $  DD0F
     CLI
     LDA # $  EE
     SEC
     SBC   $  DD04
     STA   $  65
     LDA # $  FF
     SBC   $  DD05
     STA   $  64
     LDA # $  FF
     SBC   $  DD06
     STA   $  63
     LDA # $  FF
     SBC   $  DD07
     STA   $  62
     SEC
     LDA # $  00
     LDX # $  A0
     JSR   $  BC4F
     LDA # $  93
     JSR   $  FFD2
     JSR   $  BDDD
     JMP   $  AB1E
```

brutina a cronometrar, que es EVAEXP ($ AD9E), archiconocidas sobradamente de todos los programadores, la siguiente ($ 87F7) lo transforma en dos enteros y los coloca en Y y A, lo cual ya lo tenemos listo para insertarlo en el JSR de la rutina del amigo Somoza, la cual acoplo a continuación, sin variar.

Saliendo de ella y una vez tomado el tiempo, sitúo los datos en las direcciones 62 a 65, y para poderla entrar en el fac, preparo las instrucciones de llamada llenando con 00 el acumulador y con A0 el registro X y finalmente llamando a $8C4F. Con lo cual cargo el fac con el número calculado y sólo me queda transformarlo a entero ASCII con un JSR $ BDDD e imprimirlo por la pantalla mediante JSR $ AB1E.

¡Ah! se me olvidaba: antes de presentar estos datos, borro el contenido de la pantalla, cargando el acumulador con el código ASCII CLEAR ($ 93) y llamando a $ FFD2. Esto lo hago para que no mezcle el resultado que obtengo con los posibles caracteres dibujados, fruto de la realización de la subrutina y localizarlo inmediatamente. En el fondo esto no deja de ser una FLORITURA que en según qué programa no interesa al programador, si es así, sólo con sustituir, por ejemplo el $ 93 por el $ 1D, que desplaza el cursor un lugar a la derecha, con cualquier signo ASCII que vosotros determinéis, detrás del cual se visualizará el tiempo supuesto que emplea la subrutina a cronometrar.

Otra controversia que me suscitaba la antigua subrutina era que si el resultado eran ciclos o no, o en qué norma nos salían escritos, yo creo que lo importante es que te salga un número más o menos regular ya fiable. Si te da un número es que es más rápida, sea este número lo que sea.

Por último y como también pasaba después de cronometrar varias veces la misma subrutina puede ser que contra más larga sea nos dé diferentes valores a su cronometraje, esto es debido a las interrupciones que genera el sistema. Para solventar todo esto yo suelo cronometrar una misma subrutina varias veces (dependiendo de su longitud) y apuntarme el resultado máximo y mínimo. He de comentaros también que la subrutina no es totalmente reubicable si no se modifican las direcciones del STY y STA de la cuarta y quinta línea de este programa.

En espera de que a alguien le haya servido esta modificación, o sino, la forma de realizarla, me despido cordialmente. ■

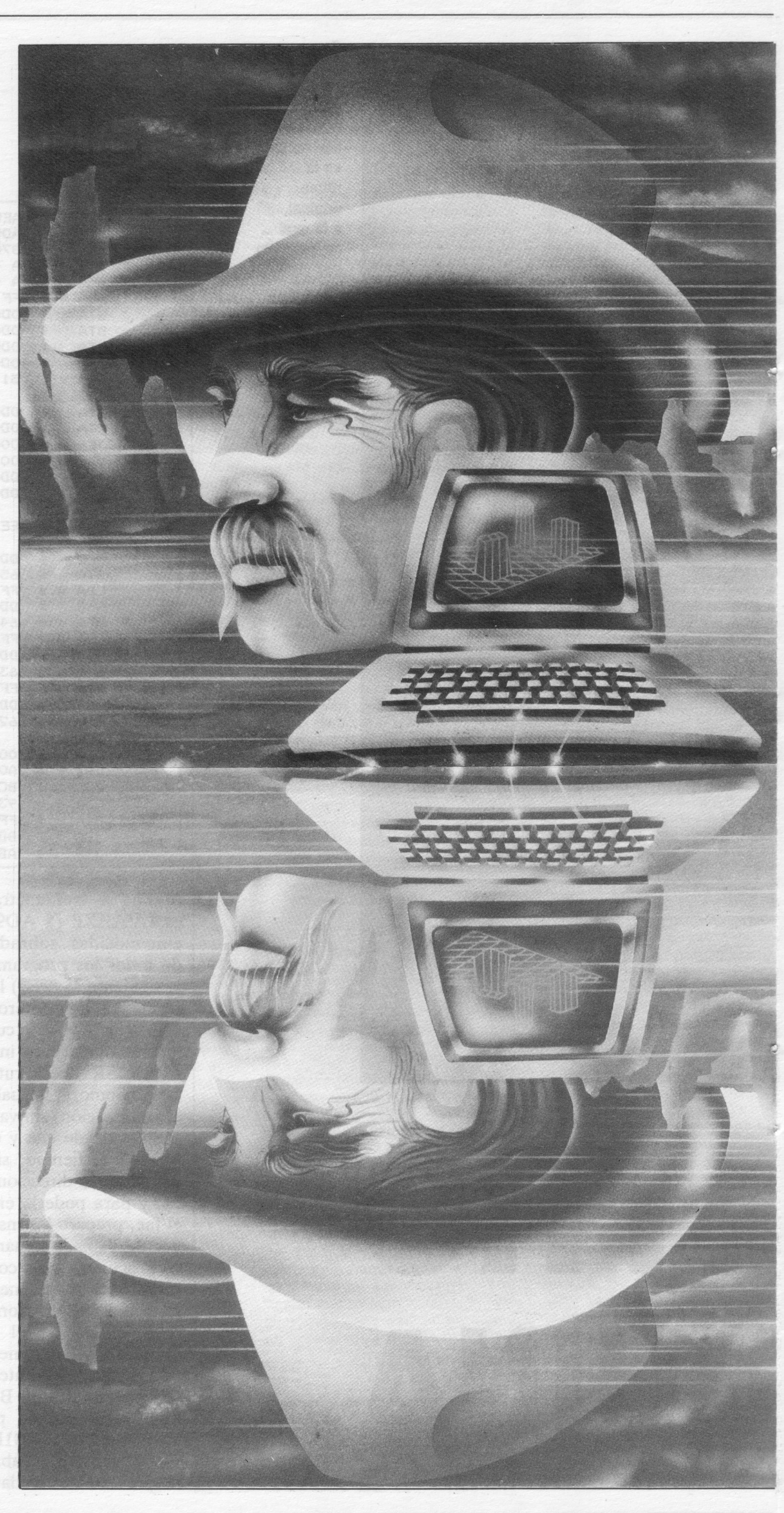
rro el contenido de la pantalla, cargando el acumulador con el código ASCII CLEAR ($ 93) y llamando a $ FFD2. Esto lo hago para que no mezcle el resultado que obtengo con los posibles caracteres dibujados, fruto de la realización de la subrutina y localizarlo inmediatamente. En el fondo esto no deja de ser una FLORITURA que en según qué programa no interesa al programador, si es así, sólo con sustituir, por ejemplo el $ 93 por el $ 1D, que desplaza el cursor un lugar a la derecha, con cualquier signo ASCII que voso- [illegible] mineis, detrás del cual se visualizará el tiempo [illegible] que emplea la sub[illegible] a cronometrar.

Otra controversia que me suscitaba la antigua subrutina era que si el resultado era [illegible] o no, o en qué norma nos salían escritos, yo creo [illegible] importante es que te salga un número más o me[illegible] regular ya fiable. Si te da [illegible] número es que es más rápida, sea este número lo que [illegible]

[illegible] por último y como también [illegible] después de cronometrar [illegible] veces la misma subrutina puede ser que contra más [illegible] sea nos dé diferentes valores a su cronometraje, esto es debido a las interrupciones que genera el sistema. Para solventar todo esto yo suelo cronometrar una misma subrutina varias veces (dependiendo de su longitud) y apuntarme el resultado máximo y mínimo. He de comentaros también que la subrutina no es totalmente reubicable si no se modifican las direcciones del STY y STA de la cuarta y quinta línea de este programa.

En espera de que a alguien le haya servido esta modificación, o sino, la forma de realizarla, me despido cordialmente.

Siguiendo con la serie de artículos sobre el juego «Sabrina», este mes se explica la forma de pasar las pantallas originales de Spectrum a Commodore.



ESPEJO 2

Por Francisco Javier Rodríguez

Los programas de este artículo también os servirán para que dichas pantallas de Spectrum puedan ser modificadas por vosotros mismos con el programa ART STUDIO, ya que los formatos de memoria son compatibles.

Otra de las posibles utilidades es la de que posteriormente las podáis utilizar como pantalla de presentación en vuestros juegos; para que esto sea posible y quede bonito deberéis poseer alguna de las versiones del freeze frame o cualquier otro tipo de copión que haga las «rayitas de color durante la carga».

Los diferentes programas son: Conversor V1.0, Cargador V1.0 y Start V1.0. El primero de los tres programas, como podréis adivinar por su nombre, es el que se encarga de pasar la pantalla de formato Spectrum a pantalla Commodore en alta resolución. Para hacer la conversión de pantalla deberéis seguir los diferentes pasos: para empezar es necesario tener el programa comercial SPECTRUM SIMULATOR con el cual tenéis que cargar vuestra pantalla de Spectrum; recordad que el Spectrum Simulator no acepta programas en c.m, por lo cual comprobad que la cinta que os pasen de Spectrum tenga la pantalla grabada como Screen$.

Una vez que tengáis la pantalla cargada (sin ningún tipo de error) grabadla en la unidad de discos como si fuera un microdrive; la sintaxis es la siguiente:

Save * «M»;1;«nombre»Screen$

Si la grabación ha tenido éxito ya podéis resetear vuestro ordenador. El siguiente paso es mediante el disco de ESPECIAL UTILIDADES y en concreto con el programa FLOPPY MASTER cambiar el formato de fichero de secuencial a programa; así como cambiarle el nombre. A partir de este momento ya tendremos preparado el fichero para trabajar con él.

Lo primero que os pedirá el programa conversor es el número de dispositivo con el que vais a trabajar (1-cinta, 8-disco); a continuación os pedirá el nombre del fichero que vais a convertir, no excediendo el nombre la longitud de 16 caracteres. A continuación el programa hará una llamada a una de las dos rutinas c.m que tiene éste programa (LOAD) la cual cargará el fichero a partir de la dirección \$4000; una vez el fichero cargado, se procederá a lo que es la propia conversión. Seguramente ya sabréis que el Spectrum divide su pantalla en tres tercios; por lo cual a la hora de tomar los datos de memoria y pokearlos en nuestra memoria de alta resolución (\$2000) también tendremos que hacerlo por tercios; consiguiendo esto con una sencilla fórmula. El proceso de transportar los datos a la memoria podéis observar que se hace un poco pesado justamente debido a que se tienen que simular los tercios.

Para convertir las pantallas se crea una matriz con la correspondencia de colores entre Spectrum y Commodore.

Una vez ya se tiene la pantalla el siguiente paso es el darle el color original; para ello se crea una pequeña matriz con la correspondencia de colores entre el Spectrum y el Commodore contenida dicha matriz en la variable CBM.

Originalmente el Spectrum simulator guarda las diferentes combinaciones de tinta y papel como si se tratara de una tabla de ejes X e Y con lo cual las diferentes combinaciones son las siguientes:

	INK							
PAPER	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
	1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7
	2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7
	3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7
	4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7
	5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7
	6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7
	7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7

Pues bien para descomponer dicho color (tinta+papel) de Spectrum y pasarlos a Commodore se usa la siguiente función:

byte = peek (22543+Y)
cero = int (byte/8) = color papel en Spectrum
uno = byte - 8* cero = color tinta en Spectrum
cero = cbm (cero) = papel en Commodore
uno = cbm (uno) = tinta en Commodore

Correspondencia de colores para variable CBM

	Commodore	Spectrum
negro	0	0
blanco	1	7
rojo	2	2
cyan	3	5
púrpura	4	3
verde	5	4
azul	6	1
amarillo	7	6

```
BYTE  DISPOSITIVO          $0B39
BYTES NOMBRE               $0B3A - $0B49
BYTE  LONGITUD NOMBRE      $0B4A

- = #$2D
# = #$23

******   RUTINA  LOAD   *****

.  0B4B   LDX $0B39           ; numero de dispositivo
.  0B4E   LDY #$00            ; dirección secundaria
.  0B50   JSR $FFBA           ; preparar dispositivo
.  0B53   LDA $0B4A           ; longitud de nombre
.  0B56   LDX #$3A            ; byte bajo dir. nombre
.  0B58   LDY #$0B            ; byte alto dir. nombre
.  0B5A   JSR $FFBD           ; poner nombre
.  0B5D   LDA #$00            ; flag de load
.  0B5F   LDX #$00            ; byte bajo dir. inicio load
.  0B61   LDY #$40            ; byte alto dir. inicio load
.  0B63   JSR $FFD5           ; carga programa
.  0B66   RTS                 ; vuelve de la rutina

*****   RUTINA  SAVE   *****

.  0B67   LDY $0B4A           ; longitud del nombre
.  0B6A   LDA #$2D            ; carácter de marca  #
.  0B6C   STA $0B39,Y         ; a último carac. nombre
.  0B6F   LDX $0B39           ; numero de dispositivo
.  0B72   JSR $FFBA           ; prepara dispositivo
.  0B75   LDA $0B4A           ; longitud del nombre
.  0B78   LDX #$3A            ; byte bajo de nombre
.  0B7A   LDY #$0B            ; byte alto de nombre
.  0B7C   JSR $FFBD           ; prepara nombre
.  0B7F   LDA #$FB            ; puntero a dir. inicial save
.  0B81   LDX #$47            ; byte bajo dir.final save
.  0B83   LDY #$3F            ; byte alto dir.final save
.  0B85   JSR $FFD8           ; graba memoria
.  0B88   RTS                 ; vuelve de rutina
.  0B89   LDY $0B4A           ; longitud de nombre
.  0B8C   LDA #$23            ; carácter de marca -
.  0B8E   STA $0B39,Y         ; a último carac. nombre
.  0B91   LDX $0B39           ; numero de dispositivo
.  0B94   JSR $FFBA           ; prepara dispositivo
.  0B97   LDA $0B4A           ; longitud nombre
.  0B9A   LDX #$3A            ; byte bajo de nombre
.  0B9C   LDY #$0B            ; byte alto de nombre
.  0B9E   JSR $FFBD           ; prepara nombre
.  0BA1   LDA #$FE            ; puntero a dir.inicial save
.  0BA3   LDX #$01            ; byte bajo dir.final save
.  0BA5   LDY #$08            ; byte alto dir.final save
.  0BA7   JSR $FFD8           ; graba memoria
.  0BAA   RTS                 ; vuelve de rutina

*****   POR AQUI SE ENTRA A RUTINA SAVE    *****

.  0BAB   LDA #$00            ; prepara puntero para
.  0BAD   STA $FB             ; dirección inicial
.  0BAF   LDA #$20            ; de pantalla en save
.  0BB1   STA $FC
.  0BB3   JSR $0B67           ; vete a save pantalla
.  0BB6   LDA #$00            ; prepara puntero para
.  0BB8   STA $FE             ; dirección inicial
.  0BBA   LDA #$04            ; de color en save
.  0BBC   STA $FF
.  0BBE   JSR $0B89           ; vete a save color
.  0BC1   LDA #$1B            ; restablece pantalla texto
.  0BC3   STA $D011           ; y memoria de color
.  0BC6   LDA #$15            ; a estado normal
.  0BC8   STA $D018
.  0BCB   RTS                 ; sal de rutina.
```

Una vez que se tiene la tinta y papel del carácter hay que saber que el commodore en la memoria de color (aquí a partir de $0400) el byte de color en Hexadecimal indica que la primera cifra (por la izquierda) indica el color de la tinta, y la segunda el color del papel por lo que si fuese 10 Hex querría decir tinta blanca sobre fondo negro.

Por lo tanto para componer el byte de color a pokear se obtiene de la siguiente forma:

col = uno*16 + cero

Y es así como después de éstas operaciones se obtiene el color final que aparecerá en pantalla.

Una vez realizadas todas las funciones de conversión el programa esperará a que pulses una tecla; si la tecla pulsada es el espacio el programa volverá a ejecutarse pero si la tecla pulsada es « <-» (flecha a la izquierda) el programa llamará a la segunda de sus rutinas en c.m, la cual se encargará de grabar la pantalla y el color ya en formato commodore, al periférico indicado al inicio del programa. La pantalla será grabada con el mismo nombre con la que se cargó el fichero original, pero sustituyendo el último carácter por un «-» y el color se grabará cambiando el último carácter por un «#»; por lo que si el periférico es,disco, podréis tener varias en un mismo diskette.

A continuación os paso a explicar las dos rutinas que incluye el programa de al lado.

Debido a la cantidad de preguntas que hán aparecido en esta revita sobre el tema de cómo podrían hacer alguna rutina para ver una pantalla de presentación en la carga de un juego y quede «bonito», también he preparado dos programas híbridos que será una pseudoforma de conseguir esto; los resultados finales serán muy parecidos a cargadores comerciales como el del juego R-TYPE cuyo cargador muestra la pantalla parando la carga del juego hasta que se pulse el espacio.

Utilizando el turbosave publicado en Commodore World se pueden obtener pantallas de carga con rayas de colores.

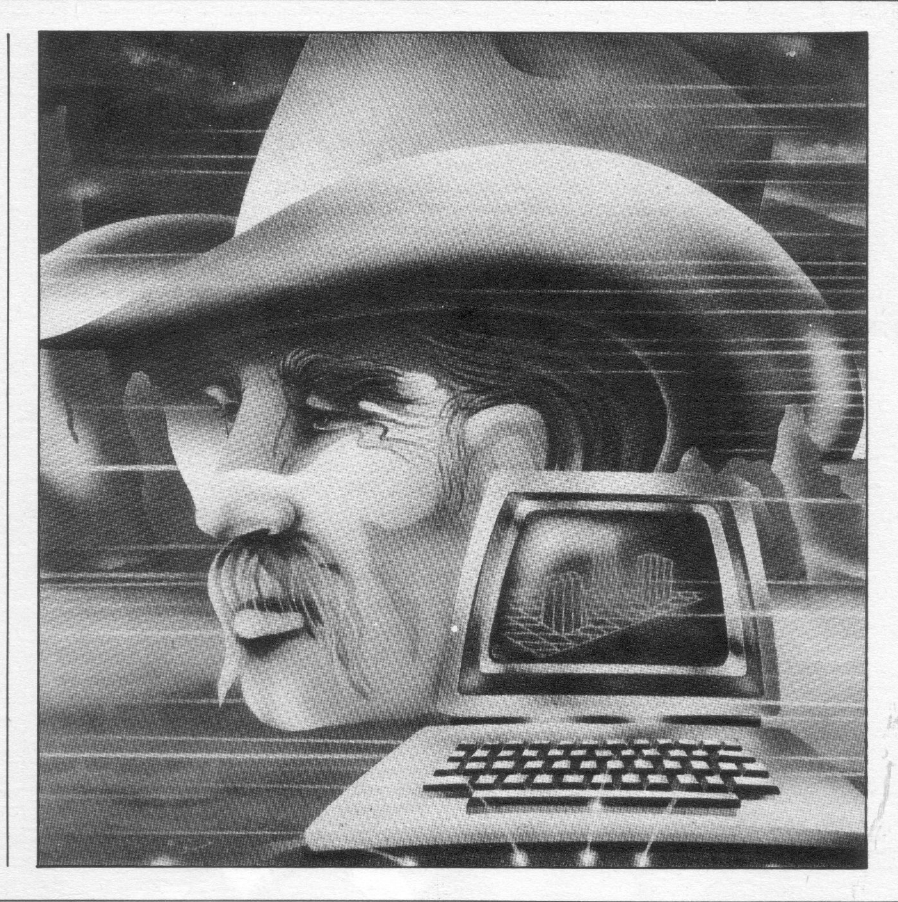
Para empezar lo primero que tendréis que hacer es cargar el turbosave (programa turbo con opción autostart) que apareció publicado en esta revista; una vez cargado, ejecutarlo con «RUN» y posteriormente borrarlo con un NEW. Seguidamente, cargar el programa START que acompaña este artículo; y una vez cargado grabarlo a cinta con la siguiente sintaxis:

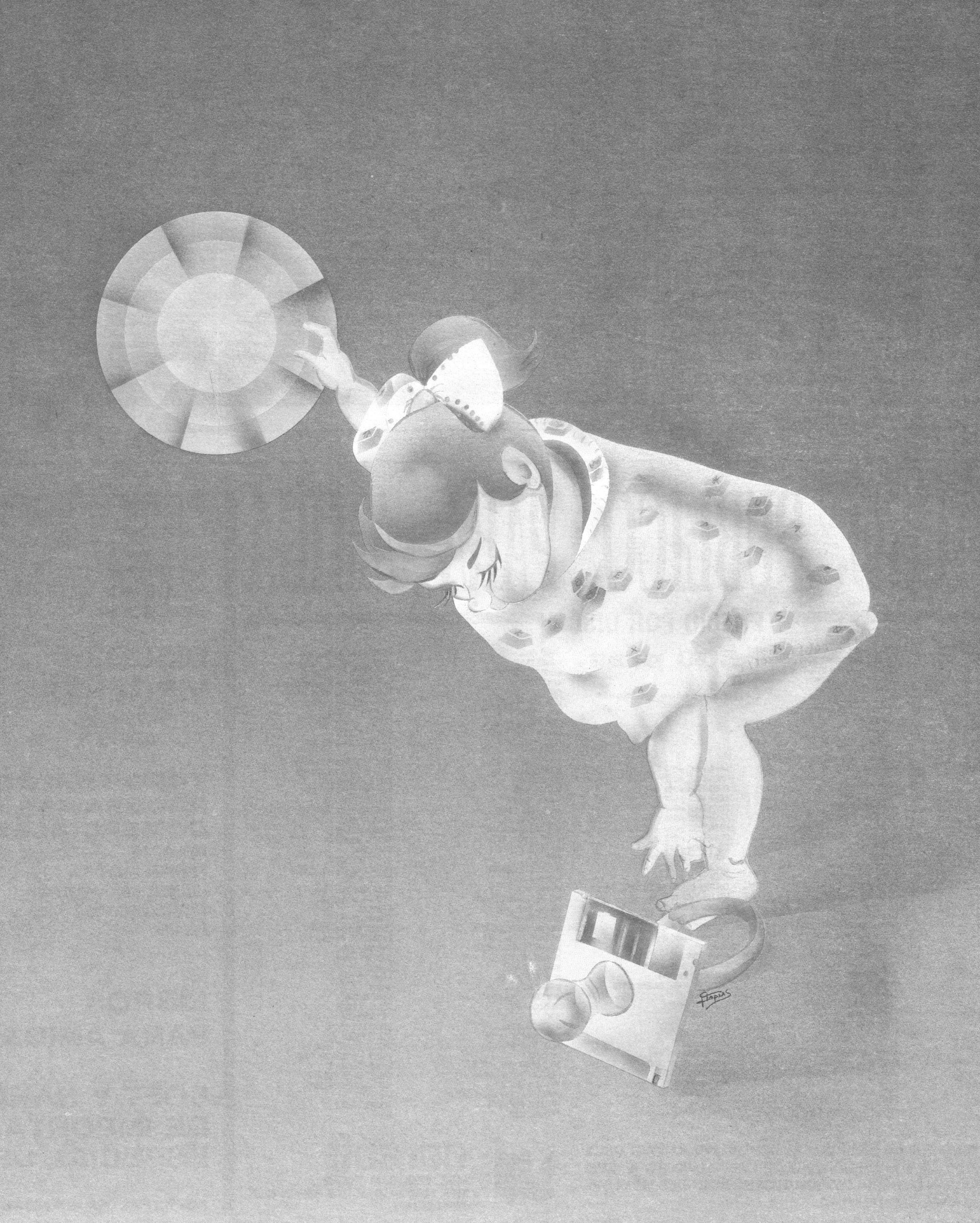
save @ «nombre»

Una vez grabado debéis apagar el ordenador e introducir el cartucho freeze-frame o cualquier otro de los que hacen aparecer rayas de colorines en el borde de la pantalla. Si usáis el freeze-frame, seleccionad la opción de configurar memoria; a continuación cargad el programa CARGADOR V1.0, cuyo listado también os adjunto, y ejecutadlo de la siguiente forma A=1:x=0: GOTO 100. A continua-

ción debéis completar las opciones que se os vayan mostrando, y una vez tengáis la pantalla deseada con su color, hay que dar al botón de turno para realizar el backup a cinta. Una vez acabado éste, hay que apagar el ordenador y a continuación volver a encenderlo otra vez con el cartucho y con la misma opción de configurar la memoria; ya sólo faltará que carguéis vuestro programa y hagáis el correspondiente back-up. Después de esto sólo tenéis que resetear el ordenador, rebobinar la cinta al principio y proceder a su carga normal. Si habéis seguido todos los pasos correctamente veréis aparecer la pantalla del juego (parándose la carga) hasta que pulséis el espacio donde se continuará con la carga del programa.

Para acabar, sólo decir que el disco Commodore World que acompaña a la revista se encuentra una pantalla de demostración para el programa CONVERSOR V1.0 que servirá posteriormente para nuestra demo jugable de «Sabrina»; también deciros que la rutina SAVE que incluye éste programa es muy fácil de modificar para que podáis grabar cualquier zona de memoria que deseéis. ■





AmigaWorld

3

ESPECIAL VIDEO

El número 3 de la revista Amiga World se pondrá a la venta a partir del próximo 15 de julio, y estará dedicado completamente al mundo del video y los efectos especiales. Es bien conocido que utilizando el Amiga se puede conseguir la calidad de un equipo profesional de video, para crear mezclas de imagen y texto, gráficos o efectos especiales. El mundo del video con el Amiga estará a tu alcance después de leer esta revista.

Un amplio artículo realizado por expertos profesionales compone la parte principal de la revista, en la que se comentan los más importantes paquetes de titulación, presentaciones, efectos especiales, digitalizadores. Programas como **TV*TEXT, TV*SHOW, PageFlipper, Deluxe Video** y muchos más estudiados en profundidad.

También contará este número con bancos de pruebas de los últimos equipos de video y genlocks para Amiga: el sistema **SATV**, los genlocks **Rendal** y **NHS, Digiview Gold 3.0...** un amplio estudio para que puedas elegir los productos que necesites.

Además, una serie de artículos sobre creación de **música** y sonido con el paquete Aegis Sonix, un cursillo de **programación en AmigaDOS** para la nueva versión 1.3 del Workbench (y banco de pruebas de la versión en castellano) y nuestras habituales secciones de **pequeños utilitarios, magia** y comentarios de los mejores juegos del momento: **Populus, Super Hang-on, Starglider II...**

La revista se completa con una amplia **GUIA DE HARDWARE,** que junto con la guía de software aparecida en el número 2 forman las listas ideales en las que localizar cualquier producto para Amiga, así como los nombres de sus distribuidores aquí en España.

Al igual que los números anteriores, la revista Amiga World 3 será una **edición limitada,** de modo que la mejor forma de conseguirla es enviando el boletín que aparece en la parte inferior de esta página, para hacer la reserva y recibirla cómodamente en tu domicilio. Simultáneamente se pondrá también a la venta el **Disco Especial 3,** con todos los programas de la revista y otros de regalo.

500 ptas.

OFERTA: REVISTA + DISCO ESPECIAL AMIGA WORLD 3, 1.995 ptas.

¡¡RESERVA TU EJEMPLAR ANTES DE QUE SE AGOTE!!

Si quieres reservar tu revista Amiga World 3 antes de que se agote, envía hoy mismo este boletín de reserva.

BOLETIN DE RESERVA - REVISTA ESPECIAL Amiga World 3

Nombre ....

Dirección ....

Población .... C.P. .... Provincia ....

Teléfono .... Modelo de Amiga ....

☐ Deseo reservar y recibir el número especial Amiga World 3 (500 ptas.).

☐ Deseo recibir el número especial Amiga World 3 junto con los discos (1.995 ptas.).

☐ Incluyo cheque por .... ptas.

☐ Envío giro número .... por .... ptas.

(Indicar el pedido también en el texto del giro)

Enviar a: Commodore World, Rafael Calvo, 18, 4.º B. 28010 Madrid.

Forma de pago: Sólo cheque o giro. No se sirven pedidos contra reembolso. Gastos de envío incluidos.

```
PROGRAMA: CONVERSOR                  LISTADO 1

1 REM COMMODORE WORLD 61                 .215
2 :                                      .234
3 REM F.JAVIER RODRIGUEZ                 .89
5 :                                      .237
100 INPUT"[CLR]DISPOSITIVO (1-8)";D      .48
110 IF D<>1 AND D<>8 THEN 100            .76
120 POKE2873,D                           .248
130 INPUT "[CLR]NOMBRE FICHERO";F$:      .46
LF=LEN(F$)
140 IF LEN(F$)>16 THEN PRINT"NOMBRE      .106
 DEMASIADO LARGO":FORT=0TO100:NEXT:
GOTO130
150 POKE 2890,LF                         .44
160 FOR T=1 TO LF:POKE2873+T,ASC(MI      .106
D$(F$,T,1)):NEXTT
170 SYS 2891                             .174
180 CBM(3)=4:CBM(4)=5:CBM(5)=3           .34
190 REM                                  .252
200 CBM(6)=7:CBM(7)=1                    .206
210 A=8192+(4*8)+(40*8):K=15             .194
220 POKE53248+17,59                      .78
230 POKE53248+24,24                      .124
240 FORI=1024TO2024:POKEI,12:NEXTI       .194
250 FORI=8192TO8192+7999:POKEI,0:NE      .252
XTI
260 FORM=0TO2                            .235
270 FORQ=0TO7                            .151
280 FORX=0TO7                            .217
290 FORR=0TO31                           .227
300 W=A+(R*8)+(X*40*8)+(M*40*8*8)        .219
310 POKEW,PEEK(16384+K):K=K+1            .239
320 NEXTR                                .221
330 NEXTX                                .23
340 A=A+1                                .167
350 NEXT Q                               .243
360 A=8192+(4*8)+(40*8)                  .73
370 NEXTM                                .231
380 FORR=0TO21                           .189
390 FORQ=0TO31                           .63
400 BYTE=PEEK(22543+Y)                   .137
410 CERO=INT(BYTE/8)                     .201
420 UNO=CBM(BYTE-(8*CERO))               .191
430 CERO=CBM(CERO)                       .225
440 COL=UNO*16+CERO                      .215
450 POKE1024+44+Q+(R*40),COL             .143
460 Y=Y+1:NEXTQ,R                        .133
470 GETA$:IFA$=""THEN470                 .125
480 IF A$=" "THEN RUN                    .229
490 IF A$="_"THEN SYS2987                .219
500 GOTO 470                             .55
```

```
PROGRAMA: CARGADOR                   LISTADO 2

100 IFA<>0ANDX=0THEN INPUT"[CLR]FIC      .50
HERO";F$
110 IF LEN(F$)>16 AND A<>0THEN100        .102
120 IFA<>0ANDX=0THEN INPUT"[CLR]DIS      .230
POSITIVO (1-8)";D
130 IF D<>1 AND D<>8 ANDA<>0THEN 12      .92
0
140 IFX=0THENPRINT"[CLR][WHT]":POKE      .220
53280,1:POKE53281,1
150 IFX=0THENP=PEEK(53248+17):P1=PE      .66
EK(53248+24)
160 POKE53248+17,59:POKE53248+24,24      .82
170 X=X+1                                .202
180 IF X=2 THEN 210                      .172
190 IF X>2 THEN 220                      .166
200 IFA<>0THENLOADF$+"-",D,1             .30
210 IFA<>0THENLOADF$+"#",D,1             .230
220 POKE192,1:POKE1,PEEK(1)OR32          .194
230 FORT=0TO1500:POKE192,1:POKE1,PE      .46
EK(1)OR32:NEXT
240 POKE53248+17,P:POKE53248+24,P1       .164
250 POKE1,PEEK(1)AND39:POKE192,0         .134
260 PRINT"[CLR][WHT]"                    .223
270 SYS 2513                             .97
```

```
ANTES DE TECLEAR EL SIGUIENTE PROGRAMA
INTRODUZCA ESTOS POKES
POKE 44,16:POKE 16*256,0:NEW
Y DESPUES, PARA GRABARLO
POKE 44,8:POKE 45,48:POKE 46,8
Y SE GRABA CON UN SAVE NORMAL
```

```
PROGRAMA: START.BASIC                LISTADO 3
1 REM COMMODORE WORLD 61                 .215
2 :                                      .234
3 REM F.JAVIER RODRIGUEZ                 .89
5 :                                      .237
10 FOR Q = 2049 TO 2097                  .122
11 READW:POKEQ,W                         .59
13 Z=Z+W                                 .223
15 NEXTQ                                 .163
17 IFZ<>4884THENPRINT"ERROR EN DATA      .99
S":STOP
100 DATA 11,8,1,0,158,50,48              .176
110 DATA 54,49,0,0,0,169,254             .136
120 DATA 141,32,208,141,33,208,169       .180
130 DATA 9,141,0,128,169,100,141         .164
140 DATA 1,128,162,1,160,1,32            .188
150 DATA 186,255,169,0,32,189,255        .212
160 DATA 32,213,255,96,0,87,208          .44
```

Sección de **JUEGOS**

AMIGA

JUANA DE ARCO

227

Basado en la historia de la famosa mujer (guerrera y santa) que acabó en la hoguera en Francia durante el Renacimiento, este programa destaca por muchas y buenas razones. La primera y más evidente es la calidad de sus gráficos. Las pantallas de fondo están digitalizadas con una calidad increíble. Después se han superpuesto los personajes que forman parte de la historia y que están animados con mucho acierto. Y para completar la labor del aspecto audiovisual, los sonidos digitalizados ambientan al máximo las escenas de lucha.

Es un juego de acción basado también en la inteligencia y sentido estratégico del jugador. Todo está unido, pero la astucia cuenta mucho. Cuando se ataca una fortaleza, al escalar las murallas de un castillo o cuando se participa en un combate en campo abierto, la rapidez de reflejos es lo principal. Sin embargo, se debe trabajar a fondo en el terreno diplomático y del espionaje, tanto como en el aspecto de definición de posiciones respecto al enemigo inglés.

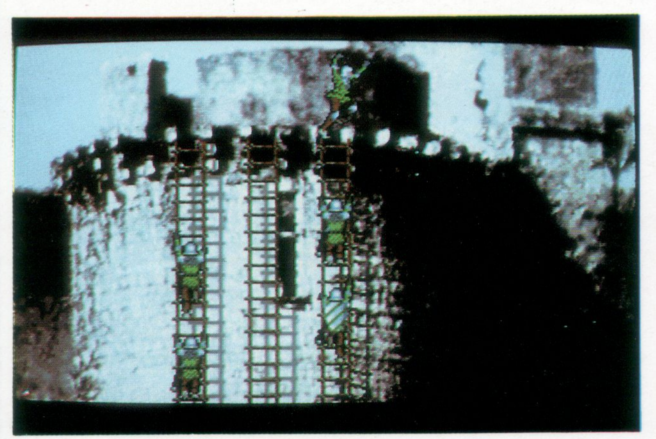
El aspecto general del programa recuerda mucho al ya antiguo Defensor de la Corona (Defender of the Crown). Este es más moderno y por lo tanto incorpora las nuevas ideas de digitalización de imágenes y sonido. El asalto a los castillos es muy emocionante, debes trepar por las escalas al mismo tiempo que te defiendes con tu escudo de las avalanchas de rocas y aceite que te lanzan los defensores desde lo alto de las almenas. Si lo consigues, habrás liberado la plaza, de lo contrario, puedes caer prisionero. Lo mismo ocurre en el caso de que tengas que defender uno de tus castillos. Tendrás que lanzar piedras o aceite contra los atacantes.

De vez en cuando tu ejército se encuentra con el de algún noble inglés que pretende conquistar más territorios para su corona. Las batallas que se plantean se llaman gloriosas y son bastante fáciles de ganar (al menos al principio, con las tropas frescas y al completo). Después de gastar la munición en cuanto a las bombardas y arqueros, tendrás pocas posibilidades contra el enemigo. Es un tema de estrategia militar el resolver las batallas con buen resultado para tus ejércitos. Debes ser rápido y preciso.

Los detalles de programación son muchos y buenos. Uno de ellos permite al usuario indicar su número de unidades de disco. Es importante ya que el programa tiene muchas pantallas y viene presentado en dos discos. Un colaborador de la revista tardó más de nueve horas en completar el juego. Teniendo en cuenta que te pueden matar y debes volver al principio, significa que nuestro joven amigo estuvo mucho más de nueve horas entretenido con el programa.

Es un juego muy bueno que combinando estrategia y la más pura acción de videojuego, te entretendrá hasta límites insospechados. ■





PROXIMO NUMERO

STORMLORD, ITALIA 90, CUSTODIAN Y MUCHO MAS.

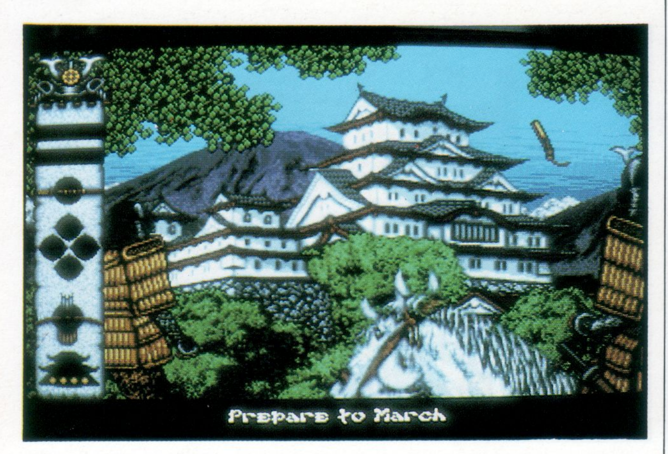
LORDS OF THE RISING SUN

228

Al estilo de los más famosos programas de estrategia y acción, pero con una calidad impresionante en cuanto a los dibujos, este juego de señores Samurais y guerreros Ninja emocionará a más de uno.

La presentación deja con la boca abierta a más de uno. Esas caras con rasgos claramente orientales, con la cabeza afeitada y recordándonos al famoso Shogun, sorprenden además por los movimientos de los ojos (un ligero parpadeo muy realista). Cinemaware, la misma compañía creadora de Capone, POW, etc., ha realizado un trabajo especialmente bueno en Lords of the Rising Sun.

Después de la impresionante presentación con sus gráficos y la música japonesa, comienza el espectáculo. Lo primero que se elige es la familia (se opta por uno de los dos nombres de patriarca familiar) con la que se combatirá. Después, al aparecer el mapa con las diferentes fortalezas, pueblos y puertos, comienzan los movimientos





estratégicos para dominar el país. Las opciones del mapa son múltiples. Desde el movimiento de los ejércitos a la visualización de la composición y disposición estratégica de cada uno de ellos. También se muestra la fuerza de que dispone cada uno, incluido el enemigo. En fin, todo lo necesario para estar en posición de combate en cualquier momento.

Según se desarrollan los acontecimientos se van produciendo situaciones curiosas. De vez en cuando el jefe del clan, el gran Samurai al que encarnas como jugador, es atacado por un guerrero Ninja. Este evento se produce mientras descansas en tu fortaleza. Con tu sable de Samurai debes esquivar las peligrosísimas estrellas que el guerrero asesino te lanza. Si no las interceptas, una de ellas acabará con tu vida y después del oportuno entierro, representado con un dibujo muy bien realizado, debes comenzar de nuevo.

Además de batallar a lo largo y ancho del Japón medieval, puedes concertar alianzas y pactos que te permitan subsistir o alcanzar con más facilidad tus objetivos. Todo forma parte de la estrategia, así que no se debe descuidar este aspecto del juego.

Cuando se lleva media hora jugando, parece que te integras en el programa. Es un juego animado por la acción de unos gráficos estupendos y también de sonidos digitalizados que lo convierten en agradable y muy entretenido. Uno de los mayores valores es la separación total respecto a los mata-marcianos.

En general todos los aspectos del programa son muy buenos. Es un juego realizado con gusto y de una calidad sobresaliente. El sello de la marca Cinemaware está presente respaldando un producto de características especiales. Es un juego para emplear horas sin aburrirse.

Debido a la gran aceptación que tienen este tipo de juegos entre nuestros lectores y amigos commodorianos, serán comentados con profusión desde estas páginas. En este mismo número se puede encontrar Juana de Arco y con números anteriores, Defender of the Crown y otros. Todos ellos tienen unas características similares en cuanto a calidad de gráficos, sonido y acción, que les confieren una especial calidad.

A pesar de su aspecto complicado, esta clase de programas permiten al usuario de Amiga descubrir rápidamente muchos aspectos del juego. La presentación es tan agradable que invita a continuar la aventura antes incluso de leer el manual. Después de la primera pantalla, se puede empezar el juego sin más complicación. ■

TIGER ROAD Y CRAZY CARS II

229

En la China de la Edad Media, innumerables leyendas se agolpaban en la memoria de las gentes, la de Lee Wong es una de las más interesantes. Cuenta cómo el alumno más aventajado del Templo de Oh Rin tiene que acabar con el reclutamiento de niños inocentes, para que así no lleguen a formar parte del malvado ejército de Ruy Ken Oh. Oh Rin dispone de numerosos poderes para lograr su objetivo: una chaqueta mágica gracias a la cual tendrá el poder de desplazarse a gran velocidad volando, y armas con las que destrozar completamente a sus enemigos.

El hacha con la que inicia la lucha así como un gran mazo que se puede obtener más o menos a la mitad de la primera pantalla, subiendo a la primera plataforma que aparece y destruyendo el primer obstáculo con que te encuentres. Durante la primera parte del recorrido, el juego resulta bastante sencillo, ya que los enemigos son continuamente destruidos a golpes de hacha o mazo. Es en la segunda pantalla donde la dificultad crece muchísimo, ya que no puedes enfrentarte a tus enemigos sino esquivarlos y escapar a la mayor velocidad posible; estos gigantes destruyen buena parte de tu fuerza con sólo tocarte. En las posteriores pantallas (muchas), hay que seguir enfrentándose a poderosos enemigos cada vez más indestructibles. El desarrollo del juego en distintos niveles, con escaleras, etc., le da una gran vistosidad y variedad.

En la parte superior de la pantalla se encuentran todos los indicadores excepto el de las vidas que te quedan que está en la parte inferior izquierda. Son seis vidas de las que inicialmente se dispone. Pero que mientras se va conociendo a los enemigos, se pierden fácilmente. El sonido es bueno y los gráficos aunque sin mucha definición en los principales participantes, no están mal.





CRAZY CARS II

Un juego de carreras, en una autopista americana y conduciendo un Ferrari F-40 (no muy conseguido), es en definitiva el esquema de este juego sin demasiadas pretensiones ni calidad.

A lo largo del juego, van apareciendo coches de policía y demás vehículos que no dejarán al F-40 seguir su loca carrera. Con sólo tocar a un coche de policía el Ferrari será destruido de una forma MUY poco espectacular o como mucho tu fabuloso coche derrapará e irá a parar a la cuneta, en donde no se puede continuar la carrera más que un pequeño recorrido o si no el vehículo explotará.

El sonido del juego es el típico en este tipo de recorridos por carreteras, frenazos, acelerones y sirenas de coche de policía. los gráficos son sencillos y las cualidades del Amiga en este aspecto no han sido en absoluto aprovechadas. Uno de los detalles es la existencia en la parte izquierda del volante del coche que se mueve al mismo tiempo que el coche y debido a la gran velocidad que éste lleva se convierte en prácticamente imposible de llevar recto por la carretera. La conducción se realiza también de noche y la carretera se divide varias veces en dos teniendo que tomar la decisión de elegir por cuál se quiere proseguir. El coche dispone de una velocidad lenta para iniciar la carrera y luego una rápida para conseguir grandes velocidades. Pulsando la tecla F2 se visualiza un mapa de las carreteras en el que se indica el inicio de la carrera y el destino así como la situación de los puestos de policía, etc. ■

Sección de JUEGOS

H.A.T.E.

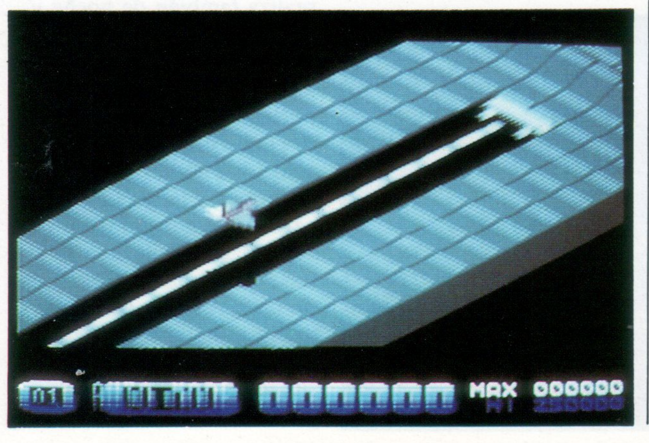
230

Hostile All Terrain Encounter, es como su nombre indica un juego bélico más. En él no encontramos nada realmente nuevo, aunque en general sí que consigue entretenernos. Manejando una nave y un vehículo de asalto terrestre, has de conseguir destruir a todos los enemigos que encuentres.

H.A.T.E. tiene 30 niveles que han de irse completando sin perder las cuatro vidas con que se inicia el juego. Una vez alcanzado el nivel diez, con el once se comienza una nueva etapa bastante más difícil; si entre los niveles once y veinte se pierden las vidas, el juego se reinicia en el once al igual que ocurre en la etapa de los niveles veinte al treinta.

En todos los niveles, se encuentran esparcidas células de plasma que se pueden conseguir una vez destruido su campo de energía y con sólo tocarlas se unen a la nave





dando así a ésta una vida extra durante el nivel en que se juega. Si la nave es alcanzada una de estas células será destruida pero no la nave. Al siguiente nivel se pasa con las vidas conseguidas durante éste. En la parte inferior están señalados los láseres disponibles para disparar en ese instante, estos láseres se recargan automáticamente siendo por tanto muy difícil no disponer de ellos cuando se necesitan.

Una vez completado con éxito el primer nivel se pasa al segundo en el cual la nave se convierte en un tanque que aunque claro, no vuela; tiene otras buenas cualidades como la posibilidad de disparar bombas y el láser a la vez, con lo cual, el poder de destrucción es bastante superior al de la nave. Al igual que la nave, el tanque también tiene que recoger las células que también a él le darán vidas suficientes como para completar el nivel. En el tercero se convierte otra vez en una nave, y así continuamente a lo largo de los treinta niveles de que se compone HATE.

Entre las cualidades interesantes de la nave, destaca la posibilidad de subir y bajar, para así poder hacer que los enemigos pasen por arriba o abajo sin tocar la nave. La nave no se choca con el suelo por mucho que bajemos, sólo lo hace contra las paredes que aparecen de frente y que convierten el escenario del juego en un pequeño túnel, una vez dentro la nave tampoco se choca contra los lados por lo que el movimiento es bastante cómodo y se puede hacer más descuidadamente.

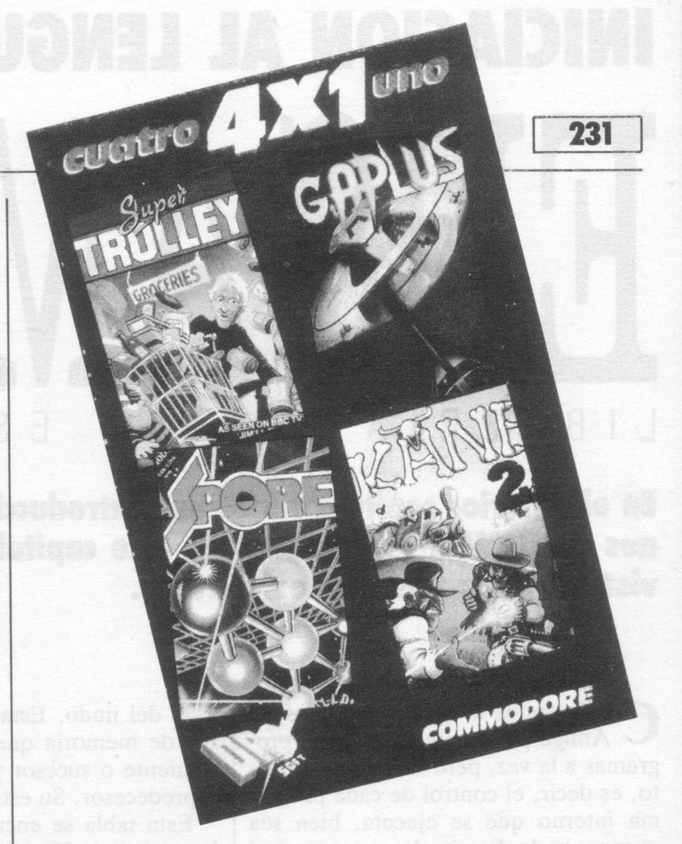
El juego se desarrolla en un scroll transversal de la esquina superior derecha de la pantalla a la inferior izquierda. En el desarrollo, H.A.T.E. nos recuerda mucho al por todos conocido, Super Zaxxon que hace ya algún tiempo hizo auténtico furor entre los incondicionales de los juegos bélicos en el C-64/128.

Tanto los gráficos como el sonido no son demasiado buenos, aunque sí es interesante la gran variedad de enemigos con los que hay que enfrentarse. El movimiento es rápido y sencillo tanto en el modo de nave como en el de tanque. El control puede hacerse con joystick y también con el teclado como en la gran mayoría de este tipo de juegos, también existe la posibilidad de dos jugadores.

La velocidad del juego, un aspecto importante que en ocasiones se descuida por la dificultad en la programación, se ha conseguido con cierta calidad en esta ocasión. Una de las características principales ya señaladas, la variedad de enemifos u objetos distintos que van apareciendo, no parece ralentizar la acción del juego. Es un buen programa. ■

Sección de JUEGOS

CUATRO X UNO



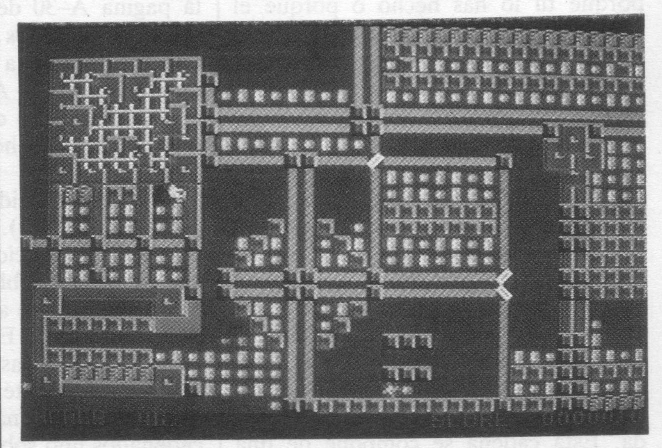
Siempre son interesantes este tipo de ofertas en las que al adquirir una cinta de juegos, en ella vienen varios. En este caso en concreto, éstos son GAPLUS, SUPER TROLLEY y en la segunda cara de la cinta, SPORE y KANE-2. Como es lógico, ninguno de los juegos tiene demasiada calidad, ni tampoco los gráficos son buenos pero resultan entretenidos en general y un buen divertimento al no centrarse demasiado en un solo tipo de juego. Por otra parte los juegos no son complicados y por tanto no necesitan de mucho tiempo para empezar a jugar, incluso sin leer las instrucciones se pueden entender enseguida.

El primero de los juegos, GAPLUS, es una especie de las conocidas moscas que todos conocemos. Se trata de ir pasando PARSECS, en cada uno de ellos han de matarse cierta cantidad de Gaplus (moscas) tras lo cual aparece el Gaplus encargado de dirigir el ataque en ese parsec. Si es eliminado, una red es tendida por tu nave, tras lo cual todos los Gaplus que en ella caigan se unirán a ti en el siguiente parsec disparando entonces a tu lado todos a la vez, de esta forma conseguirás eliminar mucho más fácilmente al resto de los Gaplus. La acción en los diferentes parsecs es muy parecida pero cada vez más complicada ya que se unen más dificultades, una de las peores es la de los cohetes que se dividen en tres o cuatro al caer. Tiene, según parece, cincuenta parsecs que recorrer… ¡Costará algún tiempo lograr el objetivo!

Es lunes, el día más tranquilo de la semana en un supermercado. SUPER TROLLEY, que es el protagonista de este juego, está en la trastienda del supermercado en que trabaja, y para empezar el jefe le manda un pequeño trabajo. Si lo termina bien tendrá más tiempo para ordenar las cosas en las estanterías del super y así el jefe no le echará. Super Trolley, para conseguir tener todo ordenado, tiene que ir con un carrito con los productos que falten desde el almacén a la tienda para reponerlos. Los mayores enemigos de Super Trolley son los niños, los perros que se cuelan y algunos recaderos que tienen la manía de descolocarlo todo. Si termina la semana con todo colocado quizá sea ascendido.

En SPORE el objetivo del juego es recuperar el antídoto para eliminar los microorganismos que se han escapado por todo el laboratorio. Para conseguirlo hay que coger todos los barriles de antídoto que hay pero que están distribuidos por todo el laboratorio.

Estás en el oeste americano, donde la ley la imponen las balas y tus disparos han de ser los más rápidos. En la primera parte, un tren a gran velocidad circula por el fondo de la pantalla, desde él diferentes bandidos disparan intentando derribarte del caballo en el que tú galopas por delante del mencionado tren. De vez en cuando aparece a los pies una roca que hay que saltar al mismo tiempo que se ha de estar atento de los disparos desde el tren. Pasada la parte del tren, aparece un fuerte que hay que atacar; desde él una gran cantidad de vaqueros que aparecen tanto en el fuerte como en sus alrededores intentarán matarte. ■





INICIACION AL LENGUAJE

ENSAMBLADO

LIBRERIA EXEC ESTRUCTURAS 3ª Part

En el anterior capítulo vimos una introducción a la librería EXEC, así como las funciones ALLOCMEM y FREEMEM. En este capítulo tendremos la oportunidad de echar un vistazo a las famosas estructuras.

Por Fernando García

Como todos muy bien sabemos, el Amiga puede ejecutar varios programas a la vez, pero cómo se hace esto, es decir, el control de cada programa interno que se ejecuta, bien sea porque tú lo has hecho o porque el Amiga así lo decidió, son mantenidos constantemente por una serie de estructuras que van encadenadas como si de una base de datos se tratara.

Entre estas estructuras se encuentran Taskings (tareas), interrupciones, librerías, periféricos, mensajes y requesters de I/O (entrada/salida). Si ponemos el ejemplo de las librerías, que es el más fácil por ahora, tienen que estar preparadas y en orden para que en cualquier momento se pueda acceder a una función. Este orden se lleva a cabo mediante las LISTAS, que es en realidad una estructura encadenada. Esta cadena se compone de una cabecera y de los llamados **nudos**, y estos nudos tienen que estar estrechamente relacionados, para que se sepa en cualquier momento dónde se encuentra lo que estamos buscando (en este caso una librería). Cada nudo apunta al siguiente y tambien al anterior como si de un gusano se tratara.

Cada nudo se divide en dos partes, la **unión** entre nudo y nudo, y el **contenido** del nudo. Esta unión son punteros de memoria que apuntan al nudo siguiente o sucesor y al nudo anterior o predecesor. Su estructura en tabla 1.

Esta tabla se encuentra también en la pagina A-30 del manual Exec, así como de los tipos de nudos, bien sea un periférico, una librería, un bloque de memoria, etc. Así es como el Amiga tiene perfecto conocimientos de lo que dispone a la hora de acceder a éstos.

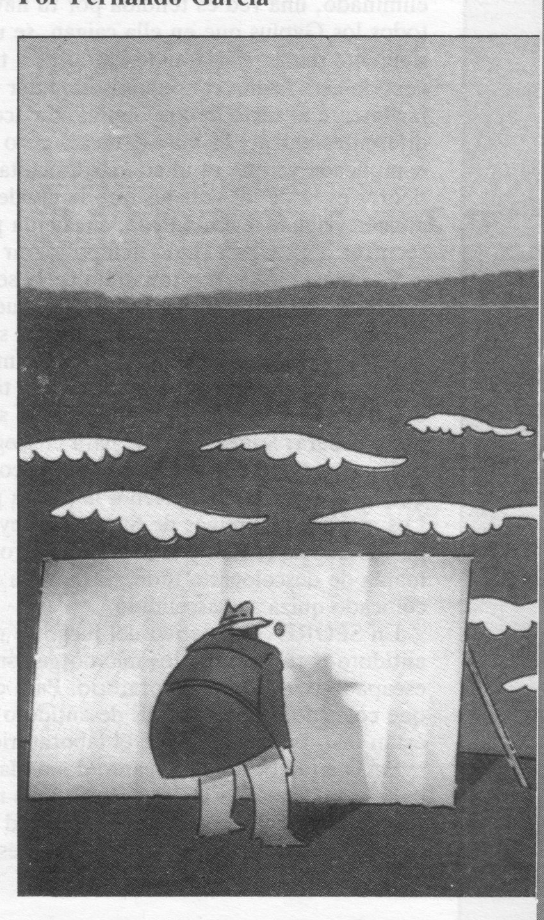
El otro contenido de las listas es la **cabecera** (header). La cabecera informa sobre la posicion del primer y ultimo nudo. (ver tabla 2)

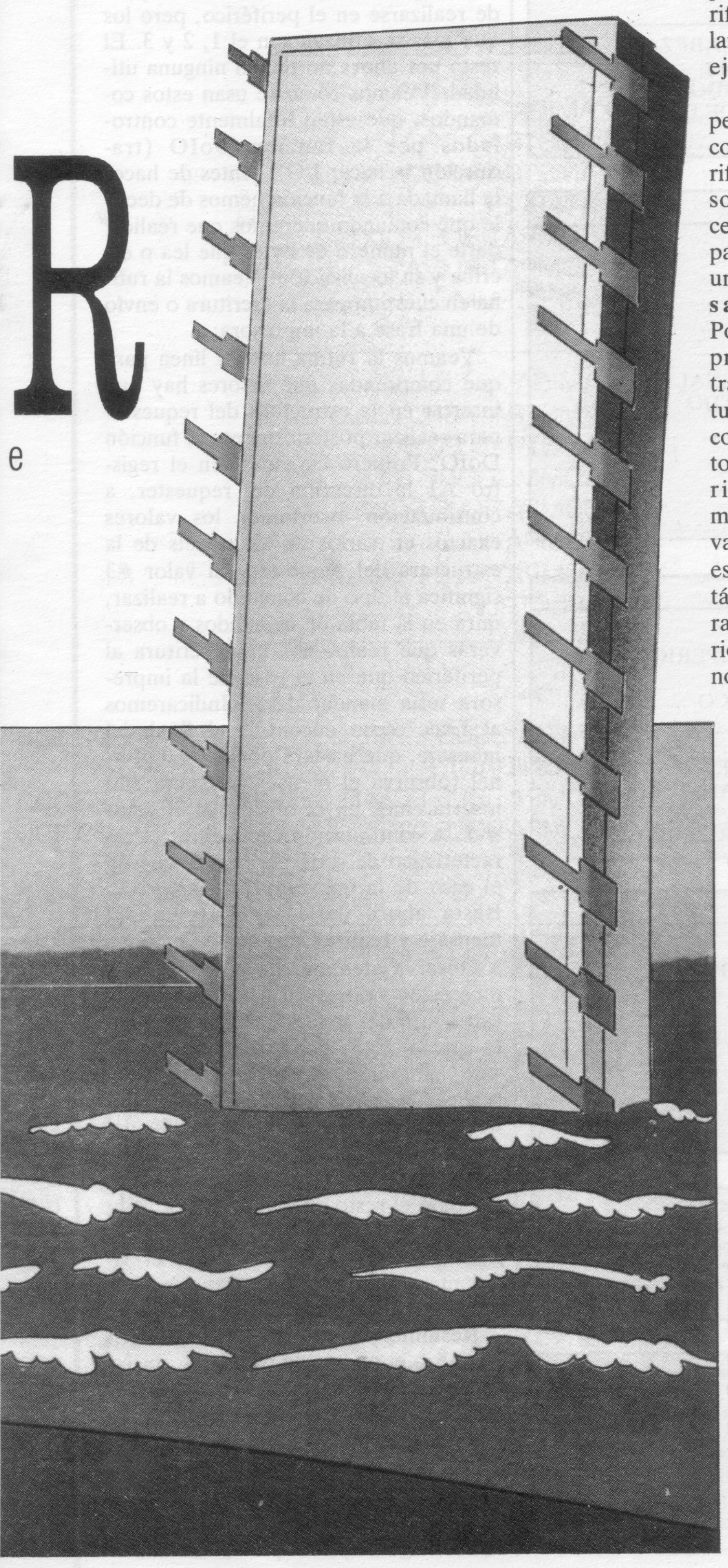
Esta estructura aparece en la página A-27 del manual Exec.

Resumiendo, las listas proporcionan al Amiga una auténtica base de datos, que constan de una serie de nudos encadenados uno sobre otro y de una cabecera que informa cuál es el primero y ultimo nudo. Estas listas pueden ser de muchos tipos, periféricos, mensajes, puertos, etc.

Acceso a los periféricos

Una de las mas importantes características del Exec es dotar a todos los





periféricos del Amiga una forma standard de entrada/salida (I/O). Esto permite comunicarse de la misma forma para todos los periféricos, bien sea impresora (printer.device), disco (trackdisk.device), serial (serial device), etc. Veamos el proceso desde el principio para acceder un periférico y más adelante pondremos un ejemplo.

Para acceder a un periférico, tomando como ejemplo el periférico de la impresora (printer.device), primero se prepara lo que se llama un **puerto de mensajes** (Message Port). Como dice la propia palabra, se trata de una estructura que el Exec reconoce como puertos para los posteriores envíos de mensajes, entre los varios task que en esos momentos están ejecutándose para dirigirse al periférico. Por ahora no nos interesa introducirnos en este tema por su enorme complejidad, así que veamos un momento la estructura correspondiente a la (tabla 3).

Veamos la rutina para poner en marcha un puerto de mensajes y observa cómo apuntan los diferentes valores a los offsets de la estructura anterior (listado 1). También recuerda que las funciones desde ahora utilizadas pertenecen a la librería Exec, que tiene que ser proviamente abierta (ver capítulo anterior).

La estructura 5anterior del puerto ocupaba $22 bytes (decimal=34) para los que se necesita reservar memoria (observa etiqueta PORT). Ahora simplemente insertamos unos valores fundamentales en la estructura, como son el tipo de puerto, exactamente el tipo 5 (puerto de mensajes), la prioridad del puerto (0=mediana) y el puntero al Task que estamos utilizando (OurTask, lo vimos en el primer capítulo). Tambien se puede poner el nombre del puerto, pero no es necesario. El resto del los offset no se ajustan, lo hace el Exec, y basta ahora ejecutar la funcion **AddPort** para que se inicialice la estructura.

El siguiente paso es inicializar el periférico, y para ello hay que fabricar otra estructura, el **Requester** (sin traducción). Se utiliza por todos los periféricos, sin embargo algunos de ellos necesitan una pequena extensión que caracteriza al periférico en cuestión, en nuestro caso el de la impresora. Esta estructura se encuentra en la pagina A-24 del manual Exec, ver tabla 4.

Esta estructura se utiliza cada vez que se quiera transmitir o recibir datos de los diversos periféricos, más tarde veremos la manera de hacerlo, y veamos la rutina para abrir un periférico (listado 2).

El primer #0 indica la unidad de impresora, en este caso la 0, pero si fuera cuestión de unidad de disco podríamos elegir entre las que tuviéramos conectadas al ordenador. El segundo #0 indica las banderas (valor sin importancia), a continuación apunta al nombre del periférico y luego a la estructura del requester que tratamos anteriormente. Para finalizar, es importante que el requester tenga conocimiento de que puerto de mensajes va a utilizar, colocandole el puntero en el offset $0E de la estructura del requester.

Todo lo explicado anteriormente sirve igualmente para utilizar otros periféricos, por ejemplo, si utilizáramos el serial bastaría cambiar los nombres de algunas etiquetas y el nombre del periférico, «printer.device» por «serial.device». Las estructuras son las mismas.

Extensiones adicionales

Comentamos anteriormente que los diversos periféricos necesitaban una extensión adicional en la estructura del REQUESTER que caracteriza al periférico. En el caso de la impresora (printer.device) hay varias extensiones según sea su utilización. Como no hay espacio para verlas, pondremos el nú-

OFFSET	ETIQUETA	CONTENIDO
$0	LN-SUCC	PUNTERO DEL NUDO SUCESOR
$4	LN-PRED	PUNTERO DEL NUDO PREDECESOR
$8	LN-TYPE	TIPO DE NUDO
$9	LN-NAME	PRIORIDAD (valores desde -128 a +127 segun prioridad)
$0A	N-SYZE	PUNTERO DEL NOMBRES

Tabla 1. Estructura de los nudos.

OFFSET	ETIQUETA	CONTENIDO
$0	LH-HEAD	PUNTERO DEL NUDO DE CABEZA
$4	LH-TAIL	PUNTERO BASE (SIEMPRE 0)
$8	LH-TALPREC	PUNTERO DEL ULTIMO NUDO DE LA LISTA
$0B	LH-TYPE	TIPO DE NUDOS DENTRO DE LA LISTA
$0C	LH-SIZE	ALINEAMIENTO (no usado)

Tabla 2. Estructura de una cabecera.

OFFSET	ETIQUETA	CONTENIDO
$0	LN-SUCC	NUDO SUCESOR
$4	LN-PRED	NUDO PREDECESOR
$8	LN-TYPE	TIPO DE NUDO
$9	LN-PRI	PRIORIDAD DEL NUDO
$0A	LN-NAME	NOMBRE DEL NUDO
$0E	MP-FLAGS	BANDERAS
$0F	MP-SIGBIT	NUMERO DEL BIT DE SENAL
$10	MP-SIGTASK	TASKING A SER SENALADO
$14	LH-HEAD	LISTA DE MENSAJES
$18	LH-TAIL	
$1C	LH-TAILPRED	
$20	LH-TYPE	
$21	LH-PAD	

Tabla 3. Estructura de un puerto de mensajes.

OFFSET	ETIQUETA	CONTENIDO
0-$12	ESTRUCTURA MN-SIZE	LISTA DE PERIFERICOS
$14	IO-DEVICE	NUDO PUNTERO DEL PERIFERICO
$18	IO-UNIT	UNIDAD
$1C	IO-COMMAND	COMANDO DEL PERIFERICO
$1E	IO-FLAGS	VARIAS BANDERAS
$1F	IO-ERROR	CODIGO DE ERROR
$20	IO-ACTUAL	ACTUAL NUMERO DE BYTES TRANSMITIDOS
$24	IO-LENGTH	NUMERO DE BYTES A TRANSFERIR
28	IO-DATA	PUNTERO DEL BLOQUE A TRANSFERIR
$2C	IO-OFFSET	OFFSET PARA VARIOS PERIFERICOS

Tabla 4. Estructura del resquester.

COMANDO	NUMERO	COMENTARIO
CMD-INVALID	0	COMANDO INVALIDO
CMD-RESET	1	INICIALIZA PERIFERICO
CMD-READ	2	LECTURA
CMD-WRITE	3	ESCRITURA
CMD-UPDATE	4	ESCRIBE TODOS LOS BUFFERS
CMD-CLEAR	5	BORRA LOS BUFFERS
CMD-STOP	6	PARO MOMENTANEO
CMD-START	7	COMIENZO DESPUES DE PARO
CMD-FLUSH	8	ABORTA PARO

Tabla 5. Mandatos de periférico.

```
            LEA       PORTPRT,A1            Apunta al puerto
            MOVE.B    #5,8(A1)              Es un puerto de mensajes
            MOVE.B #0,9(A1)                 Prioridad mediana
            MOVE.L    OurTask,$10(a1)       Apunta a nuestro puntero del task
            JSR       -$162(A6)             Funcion addport del Exec
PORTPRT     DS.B      $22
```

Listado 1.

```
            MOVE.L    #0,D0                 Unidad=0 impresora
            MOVE.L    #0,D1                 Banderas
            LEA       PRTNAME,A0            Apunta al nombre del periferico
            LEA       REQ,A1                Apunta al bloque requester
            MOVE.L    #PORTPRT,$E(A1)       Salva puerto (puerto de mensajes)
            JSR       -$1BC(A6)             Funcion opendevice
PRTNAME     DC.B      'printer.device';0
REQ         DS.B      $30
```

Listado 2.

mero de bytes que ocupa esa extensión a la estructura.

Ahora que ya hemos inicializado el periférico podremos mandarle los diversos comandos que puede realizar. Estos comandos vienen numerados de 0 a 8, ver tabla 5.

Cualquiera de estos comandos puede realizarse en el periférico, pero los que más se utilizan son el 1, 2 y 3. El resto por ahora no tienen ninguna utilidad. Veamos cómo se usan estos comandos, que estan totalmente controlados por la funcion DoIO (traducción = hacer I/O). Antes de hacer la llamada a la función hemos de decirle qué comando queremos que realice, darle el número de bytes que lea o escriba y su localización. Veamos la rutina en cuestion para la escritura o envío de una frase a la impresora:

Veamos la rutina línea a línea para que comprendas qué valores hay que insertar en la estructura del requester para realizar posteriormente la función DoIO. Primero cargamos en el registro A1 la dirección del requester, a continuación insertamos los valores exactos en varios de los offsets de la estructura del requester. El valor #3 significa el tipo de comando a realizar, mira en la tabla de comandos y observarás que realizamos una escritura al periférico que en el caso de la impresora sería mandar datos. Indicaremos al Exec cómo encontrar el final del mensaje, que bastará poner un 0 al final (observa el mensaje), y para ello insertaremos en el offset $24 el valor #-1. A continuación viene el offset característico de cada periférico, que en el caso de la impresora es siempre 0. Basta ahora darle la dirección del mensaje y realizar la función DoIO.

Observa atentamente cómo insertamos estos valores en los offsets de la estructura del requester. Ten en cuenta que el Exec debe saber qué mensaje, contenido, dirección, y final para realizar la función. Por ejemplo, la línea donde se inserta la dirección del mensaje compárala con el offset $28 del requester (IO–DATA), haz lo mismo para el resto de los offsets. Si ya lo has comprendido unamos todas las rutinas necesarias y nos saldría el siguiente programa que ya puedes ensamblar y ejecutar:

Resumiendo el programa: primero inicializa el puerto de mensajes, inicializa el periférico, envía el mensaje y una vez enviado cierra el periférico y el puerto. Este proceso es fundamental e indispensable para realizar cualquier modo de acceso a los periféricos. Si sigues investigando por tu cuenta puedes averiguar qué se puede leer desde

LO USAN LAS TELEVISIONES DE TODO EL MUNDO

GENLOCK *PAL*



El video interface de *AMIGA*
que puede usarse con todos los modelos
(*AMIGA* 500-1000-2000)
Este aparato funciona con cualquier señal de video, magnetoscopio y discos laser
Superpone gráficos de *AMIGA* con señales de video.
Puede usarse con monitores de video de bajo precio

BC

BARNACOMPUTER

BARNACOMPUTER, S.A. Mallorca, 218 - Tel. (93) 254 63 02 - 254 33 09 - Telefax (93) 254 5623 - 08008 Barcelona

```
                LEA         REQ,A1                        Apunta al requester
                MOVE.W      #3,$1C(A1)                    Efectua comando cmd_write
                MOVE.L      #-1,$24(A1)                   Para si encuentra 0 (fin mensaje)
                MOVE.L      #0,$2C(A1)                    Offset ,no en printer.Device
                MOVE.L      #MSG,$28(A1)                  Direccion del mensaje
                JSR         -$1C8(A6)                     Funcion DoIO
MSG             DC.B        'ESTO ES UN MENSAJE',$0A,0    0=Fin mensaje
```

Listado 3.

```
                XREF        OurTask
                XREF        _SysBase
                XDEF        START

******** ABRE PUERTO DE MENSAJES
START           LEA         PORTPRT,A1                    Apunta al puerto
                MOVE.B      #5,8(A1)                      Es un puerto de mensajes
                MOVE.B      #0,9(A1)                      Prioridad mediana
                MOVE.L      OurTask,$10(a1)               Apunta a nuestro puntero del task
                JSR         -$162(A6)                     Funcion addport del exec

******** ABRE PERIFERICO
                MOVE.L      #0,D0                         Unidad=0 impresora
                MOVE.L      #0,D1                         Banderas
                LEA         PRTNAME,A0                    Apunta al nombre del periferico
                LEA         REQ,A1                        Apunta al bloque requester
                MOVE.L      #PORTPRT,$E(A1)               Salva puerto (puerto de mensajes)
                JSR         -$1BC(A6)                     Funcion opendevice

******** COMPRUEBA SI HAY ERROR
                TST.L       D0                            Comprueba si ha abierto
                BNE         ERROR                         Si no, salida

******** MANDA MENSAJE A IMPRESORA
                LEA         REQ,A1                        Apunta al requester
                MOVE.W      #3,$1C(A1)                    Efectua comando cmd_write
                MOVE.L      #-1,$24(A1)                   Para si encuentra 0 (fin mensaje)
                MOVE.L      #0,$2C(A1)                    Offset ,no en printer.Device
                MOVE.L      #MSG,$28(A1)                  Direccion del mensaje
                JSR         -$1C8(A6)                     Funcion DoIO

******** CIERRA
                LEA         REQ,A1                        Apunta al requester
                JSR         -$1C2(A6)                     Cierra periferico
ERROR           LEA         PORTPRT,A1                    Apunta al puerto de mensajes
                JSR         -$168(A6)                     Quita puerto
                RTS                                       Fin

******* STRUCTURAS
PRTNAME         DC.B        'printer.device',0
MSG             DC.B        'ESTO ES UN MENSAJE',$0A,0    0=Fin mensaje
                CNOP        0,2
REQ             DS.B        $4E                           Estructura requester
PORTPRT         DS.B        $22                           Estructura puerto
                END
```

Listado 4.

un modem, e incluso averiaguar si hay papel preparado en la impresora (por falta de espacio te dejamos este trabajo a ti). Otra de las opciones importantísimas es el de poder leer un bloque de un disco en memoria para que puedas investigar, pero antes veremos otras cosas acerca de la librería Exec.

Librería EXEC

He aquí un resumen de las funciones más importantes de la librería Exec. Es este cuadro tendrás a tu disposicion los offsets, valores que se han de insertar en los registros y la explicación de los mismos.

FORBID

Estructura: forbid().
Descripción: Esta función se usa cuando un programa (task) va a acceder a estructuras que también están siendo usadas por otros programas, para prever posibles errores. Por ejemplo, cuando se va a acceder a listas de bloques de memoria, ya que pueden ser usados por otros programas. En resumen, es una parada de multitasking.

PERMIT

Estructura:
permit().
Descripcion:
Lo contrario a la funcion forbid, restablece el multitasking.

ALLOCMEM

Estructura:
Bloque memoria = AllocMem (bytes, requerimientos).
Registros:
D0 D0 D1.
Parámetros:
Bytes = cantidad de memoria a reservar requerimientos = parámetros de memoria (FAST MEMORY-,CHIP MEM.).
Resultado:
El puntero o dirección del bloque de memoria.
Descripción:
Permite reservar un espacio de memoria para su posterior utilización. Los requerimientos permiten dar a conocer qué tipo de memoria se va a reservar. Los valores exactos se dieron el el capítulo del mes pasado.

FREEMEM

Estructura:
FreeMem (puntero del bloque, numero de bytes).
Registros:
A1 D0.
Parámetros:
Puntero = dirección del bloque obtenido.
Número de bytes = cantidad de memoria a desocupar.



Es importante que el requester tenga conocimiento de que puerto de mensajes va a utilizar, colocándole el puntero en el offset $0E.

Descripción:
Desocupa el bloque de memoria reservado con la funcion Allocmem.

ADDPORT

Estructura:
AddPort (port).
Registros:
A1.
Parámetros:
Port = puntero de estructura del puerto.
Descripción:
Permite añadir un puerto de mensajes a la lista de mensajes del sistema. Antes de ejecutar la función han de insertarse ciertos valores (prioridad, tipo de puerto) en la estructura.

REMPORT

Estructura:
RemPort (port).
Registros:
A1.
Parámentros:
Port = puntero de la estructura del puerto.
Descripción:
Cierra un puerto de la lista de puertos.



Otra de las opciones importantísimas es el de poder lee un bloque de un disco en memoria para que puedas investigar.

OPENLIBRARY

Estructura:
Base librería = OldOpenLibrary (nombre).
Registros:
A1.
Parámetros:
Nombre = nombre de la librería terminado con 0.
Resultado:
Se obtiene la base de la librería.
Descrición:
La función permite abrir una librería retornando una vez ejecutada con la base de la librería. Si retorna nulo en el registro D0 ha ocurrido un error.

CLOSELIBRARY

Estructura:
CloseLibrary (nombre).
Registros:
A1.
Parámetros:
Nombre = nombre de la librería.
Descripción:
Permite cerrar una librería.

OPENDEVICE

Estructura:
error = OpenDevice (nombre, unidad, requester, banderas).

```
                XREF        OurTask
                XREF        _SysBase
                XREF        _DOSBase
                XDEF        START
                XREF        _stdin
                XREF        _stdout
******** OCUPA MEMORIA PARA BUFFER, PUERTO Y REQUESTER
STAR            MOVE.L      #$1600,D0               numero de bytes para buffer de pista
                MOVE.L      #$10002,D1              parametros CHIP_MEMORY
                JSR         -$C6(A6)                inicializa bloque memoria
                MOVE.L      D0,DISKBLOCK            salva puntero memoria
                MOVE.L      #$4E,D0                 bytes para requester
                MOVE.L      #$10002,D1              CHIP_MEMORY
                JSR         -$C6(A6)                inicializa bloque
                MOVE.L      D0,DISKIOREQ            salva puntero
                MOVE.L      #$22,D0                 bytes para puerto
                MOVE.L      #$10002,D1              CHIP_MEMORY
                JSR         -$C6(A6)                inicializa bloque
                MOVE.L      D0,DISKPORT             salva puntero
******** PREPARA PUERTO DE MENSAJES
                MOVE.L      DISKPORT,A1             get a pointer
                MOVE.B      #5,8(A1)                tipo  de puerto= mensajes
                MOVE.B      #0,9(A1)                prioridad mediana
                MOVE.L      OurTask,$10(A1)         salva puntero del task
                JSR         -$162(A6)               inicializa puerto
******** ABRE PERIFERICO TRACKDISK.DEVICE
                MOVE.L      #0,D0                   unidad 1. Drive DF0:
                MOVE.L      #0,D1                   sin banderas
                LEA         TDNAME,A0               nombre del periferico
                MOVE.L      DISKIOREQ,A1            apunta a la estructura del requester
                MOVE.L      DISKPORT,$0E(A1)  lo salva en estructura del requester
                JSR         -$1BC(A6)               abre periferico trackdisk.device
******* RUTINA CHEQUEO DE PISTAS
LOOP1           MOVE.L      DISKIOREQ,A1            apunta requester
                MOVE.W      #2,$1C(A1)              comandoREAD
                MOVE.L      #$1600,$24(A1)          numero de bytes
                MOVE.L      DISKBLOCK,$28(A1) apunta al bloque para lectura
                CLR.L       D0                      borra registro
                MOVE.B      TRACK,D0                carga pista
                MULU        #$1600,D0               calcula offset de pista
                MOVE.L      D0,$2C(A1)              inserta en offset
                JSR         -$1C8(A6)               ejecuta funcion DoIO
******** COMPRUEBA SI HAY ERROR AL LEER PISTA
                MOVE.L      DISKIOREQ,A1            apunta requester
                TST.B       $1F(A1)                 comprueba offset $1f= errores
                BEQ.S       TRACK_OK                salta si no hay error
                BRA         ERROR                   ir salida mensaje error
TRACK_OK        ADDQ.B      #1,TRACK                anadir pista
                CMP.B       #160,TRACK              comprobar si es pista 80 (160/2 head)
                BNE         LOOP1                   si no realiza de nuevo el proceso
******** TERMINA CHEQUEO, APAGA MOTOR DRIVE
                MOVE.L      DISKIOREQ,A1            apunta requester
                MOVE.W      #9,$1C(A1)              commando TD_MOTOR
                MOVE.L      #0,$24(A1)              longitud nula
                MOVE.L      #0,$28(A1)              buffer nulo
                MOVE.L      #0,$2C(A1)              offset 0
                JSR         -$1C8(A6)               realiza comando apagado motor
******** CIERRA TRACKDISK.DEVICE
                MOVE.L      DISKIOREQ,A1            apunta requester
                JSR         -$1C2(A6)               cierra periferico
******** CIERRA PUERTO DE MENSAJES
                MOVE.L      DISKPORT,A1             apunta al puerto de mensajes
                JSR         -$168(A6)               quita puerto
******** DESOCUPA BLOQUES DE MEMORIA
                MOVE.L      #$1600,D0               buffer de la pista
                MOVE.L      DISKBLOCK,A1            carga puntero
                JSR         -$D2(A6)                cierra bloque memoria
                MOVE.L      #$4E,D0                 requester
                MOVE.L      DISKIOREQ,A1            carga puntero
                JSR         -$D2(A6)                cierra bloque
                MOVE.L      #$22,D0                 puerto
                MOVE.L      DISKPORT,A1             carga puntero
                JSR         -$D2(A6)                cierra bloque
******* SALIDA DE MENSAJES
                MOVE.L      _DOSBase,A6             apunta dos.library
                MOVE.L      #MSG2,D2                mensaje 2
                MOVE.L      #MSG2E,D3               fin de mensaje
                SUB.L       D2,D3                   calcula longitud mensaje
                BRA         PRINT                   imprime
                RTS                                 retorna cli
ERROR           MOVE.L      _DOSBase,A6             apunta dos.library
                MOVE.L      #MSG1,D2                mensaje 1
                MOVE.L      #MSG1E,D3               fin de mensaje
                SUB.L       D2,D3                   calcula longitud
PRINT           MOVE.L      _stdin,D1               obtiene a donde va el mensaje (CLI)
                JSR         -$30(A6)                escribe
                RTS                                 retorna a Cli
```

Listado 5.

Registros:
A0 D0 A1 D1

Parámetros:
Nombre = nombre del periférico unidad = número de unidad requester = estructura del requester (ya vista) banderas = varias banderas (generalmente 0).

Resultado:
Si retorna la función diferente de nulo, es porque ha ocurrido un error.

Descripción:
Permite abrir un periférico e inicializar el requester correspondiente.

CLOSEDEVICE

Estructura:
CloseDevice (requester).

Registros:
A1.

Parámetros:
Requester = estructura del requester con el que ha sido abierto.

Descripción:
Con esta funcion se informa al sistema la finalizacion del acceso a un periférico.

DOIO

Estructura:
Error = DoIO (requester).

Registros:
A1

Parámetros:
Requester= estructura del requester.

Resultado:
Retorna nulo si hay error.

Descripción:
Ejecuta un comando especificado en el requester del periférico. No retornará la función a menos que se haya completado el comando propuesto. Tienes que tener en cuenta que si mandas un comando mal formado tu programa puede quedarse 'colgado'.

Ejemplo práctico

Estos son los comandos fundamentales para manejar superficialmente la librería Exec con sus características. Ahora vamos a proponerte una pequena utilidad, que es un chequeador de errores para el drive. Resumiendo, el programa carga pista tras pista para comprobar si hay errores. Así tendrás la oportunidad de comprobar las diferencias que hay entre el modo de acceso del periférico anterior (printer.device) y el periférico del drive (trackdisk.device). El manejo del programa es sencillo, copia el programa en RAM:,

NOVEDADES abc analog, s. a.
PARA ORDENADORES AMIGA

GVP

GREAT VALLEY PRODUCTS

A2000 - HARD CARD

Discos duros en tarjeta para AMIGA 2000

Con fantásticas características, **ejemplo: A2000-HARD CARD 80Q** con zócalo para EPROM de autoarranque v 1.3; DMA; 80 MB de capacidad; **64 Kb de memoria CACHE; 19/11 mseg. de tiempo de acceso** y 2MB/seg. de velocidad de transferencias de datos; controlador SCSI para otros periféricos, etc.

TARJETAS CONTROLADORAS SCCI/RAM MULTIFUNCION

Con controlador SCSI de disco duro y posibilidad de instalar 2MB DRAM, zócalos para EPROMS de autoarranque del A2000, velocidad de hasta 4MB/seg. conector externo SCSI para hasta 7 periféricos en cascada, etc.

DISCO DURO REMOVIBLE PARA A2000

44 MB de capacidad, 25 mseg. de acceso, tecnología Winchester, formato 5,25", 1/2 altura, 1,25 MB/seg., buffer de 8KB, etc. Incluye la EPROM de autoarranque.

PARA AMIGA 2000

— DISCOS DUROS DE 20 Y 40 MB
— TARJETAS PARA AUTOARRANQUE
— TARJETAS CON 2 A 8 MB DE MEMORIA

* * * OTRAS NOVEDADES * * *

— TARJETAS ACELERADORAS CMI
— EMULADOR A-MAX DE MACINTOSH
— MODEMS SUPRA 2400 BPS
— DE LUXE PAINT 3
— PAGE RENDER 3D-PAL
— CALIGARI CONSUMER

Distribuido por **abc analog, s. a.**

Santa Cruz de Marcenado, 31
28015 MADRID Tels. (91) 248 82 13

Télex: 44561 BAB CE
Fax: (341) 542 50 59

RUEGO ENVIEN INFORMACION SIN COMPROMISO A:

NOMBRE

DIRECCION

TELEFONO

ORDENADOR

MAYO 89

```
******** MENSAJES
MSG1            DC.B    'ERROR EN DISCO',$0A
MSG1E           DC.B    0
MSG2            DC.B    'NO HAY ERRORES',$0A
MSG2E           DC.B    0
******** PERIFERICO
TDNAME          DC.B    'trackdisk.device',0
******** PUNTEROS
DISKIOREQ       DC.L    0
DISKPORT        DC.L    0
DISKBLOCK       DC.L    0
TRACK           DC.B    0
                END
```

Listado 5 (continuación)..

```
        MOVE.L  DISKIOREQ,A1                    apunta requester
        MOVE.W  #2,$1C(A1)                      comando READ
        MOVE.L  #512,$24(A1)                    numero de bytes
        MOVE.L  DISKBLOCK,$28(A1) apunta al bloque para lectura
        MOVE.L  #$38C00,$2C(A1)                 inserta valor en offset
        JSR     -$1C8(A6)                       ejecuta funcion DoIO
```

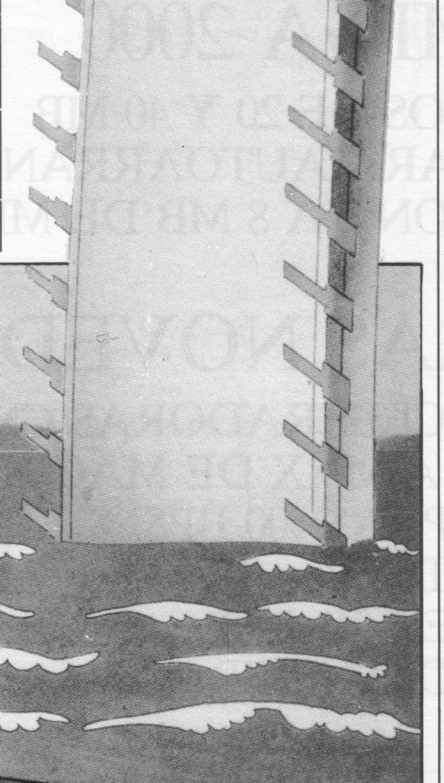
Listado 6.

inserta un disco en DF0: y ejecuta el programa, espera que compruebe los errores y el aviso en pantalla de que ha terminado el proceso.

Si te has fijado bien, muchas de las partes de este programa coinciden con las rutinas que vimos al principio del capítulo. Primero inicializa los bloques de memoria correspondientes al requester, al puerto de mensajes y al buffer que contendrá los datos de las pistas. A continuación se abre el puerto de mensajes, que aquí no va a tener una función importante, y para ello se insertan valores fundamentales como son el tipo de puerto, su prioridad y desde qué task o programa va a ser utilizado.

Posteriormente abre el periférico colocando en los offsets indicados la unidad de disco a la que se va a acceder, el nombre del periférico y el puerto de mensajes. Como podrás observar, prácticamente son los mismos parámetros que vimos antes, y solamente cambia el nombre del periférico.

Ahora es muy simple la lectura de los sectores para comprobar si no contienen errores. Veamos los valores exactos que te servirán posteriormente para acceder a cualquier pista del drive:



1. Primero el comando, y como pretendemos acceder utilizaremos el valor 2 (READ) correspondiente a la tabla de comandos anteriormente vista y lo pondremos en el offset $1C de la estructura del requester.

2. Luego insertaremos el número de bytes a leer partir del bloque que queramos acceder. Un sector entero ocupa 11 bloques y cada bloque 512 bytes y multiplicando nos da 5632 ($1600 hexadecimal). Utilizaremos el offset $24 del requester.

3. Apuntaremos el bloque que inicializamos al principio para que el Amiga los inserte correctamente.

4. Ahora viene lo mas importante, calcular lo que se llama el offset del bloque del disco, es decir, a qué bloque va a acceder el Amiga. Y ésta es la fórmula para conocerlo:

offset= 512 * (sector + 11 * cabeza + 11 * 2 * cilindro)

Como sabes el drive del Amiga consiste en 2 cabezas, 80 cilindros, y 11 sectores por cilindro. Por lo tanto para averiguar el offset correspondiente al sector 0 del cilindro 3 y cabeza 0 insertaríamos los valores correctos y hacemos la operación, dando en total en hexadecimal $8400. Esto el programa está simplificado, pues los valores del sector y la cabeza son constantes, realizando una simple multiplicación del valor $1600 por el cilindro, en este caso 3 nos da el mismo resultado.

Si quieres modificar el programa para que te lea sólo un bloque tienes que dar todos los valores correctos e insertarlos en el offset $2C antes de realizar la función. También tienes que tener en cuenta que el número de bytes cambia, pasando de $1600 de un cilindro entero a #512 de que consta un bloque. Este sería un ejemplo para el sector 3, cabeza 1 y cilindro 20 (valor de offse=$38C00):

Cada vez que lee una pista comprueba si hay algún error en el offset $1F del requester (mira la estructura del requester). Por falta de espacio no podemos poner el tipo de error producido, simplemente observa cómo lo hace.

El proceso de lectura de pistas se repite hasta llegar al valor 160 que en realidad es 80 cilindros por el número de cabezas del drive. Una vez llegado a ese valor concluye el chequeo del drive, ahora sólo queda cerrar todo lo abierto, los bloques de memoria, etc. Con esto es suficiente para observar el proceso tan igualitario para acceder a los periféricos, y las sutiles diferencias que existen. Esto le hace al Amiga una potente herramienta de comunicación con el exterior y con su interior. ■



Ya está a la venta el quinto disco Amiga World, con los programas que periódicamente publicamos en la sección para Amiga dentro de Commodore World.

Este disco corresponde a los números 58, 59 y 60 de la revista. Como en otras ocasiones, incluye todos los programas que han aparecido listados, así como otros de regalo. Como programa especial se incluye la ampliación **Gadgets Basic**, un magnífico programa que puede ser de gran utilidad para todos los usuarios que emplean el Amiga Basic como lenguaje de programación y además quieren hacer uso de las «intuitivas» posibilidades del Amiga. El disco incluye también los últimos listados del **Curso de C**, los del **Curso de Ensamblador** y otros programas de demostración y utilizades.

POR SOLO 1.700 PESETAS, EL DISCO CON TODOS LOS PROGRAMAS DE LAS REVISTAS 58, 59 Y 60.

BOLETIN DE PEDIDO - DISCOS AMIGAWORLD V

Nombre ....

Dirección ....

Población .... C.P. .... Provincia ....

Teléfono .... Modelo de Amiga

☐ Deseo recibir el disco AMIGAWorld 5 (1.700 ptas.)

☐ Deseo suscribirme por un año (4 discos y 11 revistas) al precio especial de 9.000 ptas., a partir de la revista número ....

☐ Incluyo cheque por .... ptas.

☐ Incluyo giro número .... por .... ptas.

Enviar a: COMMODORE WORLD, Rafael Calvo, 18, 4.º B. 28010 Madrid.

Forma de pago: Sólo cheque giro. No se sirven pedidos contrarreembolso. Gastos de envío incluidos.

La suscripción se puede pagar por tarjeta VISA o MASTERCARD, bien por carta o por teléfono.

Dominio Público

Esta sección está dedicada a breves comentarios de aquellos programas de dominio público que puedan ser de especial interés para todos los usuarios de Commodore.

AMIGA

BCS-AMIGA 4

El disco de este mes es verdaderamente antiguo (más o menos de mediados de 1986) pero incluye algunos programas muy interesantes. Entre ellos hay algunos listados en Basic, en C y el fenomenal juego **Hack,** un auténtico clásico. Como la mayoría de los discos de dominio público, no autoí arranca, de modo que hay que insertar en primer lugar un disco con Workbench para poder investigar en su interior. Para ejecutar los programas son necesarios unos mínimos conocimientos de AmigaDOS.

Hay cuatro grandes áreas en este disco: Basic, C, CLI y Juegos. En cada uno de ellos se encuentran diversos programas y documentación sobre todos ellos.

Programas en Basic utilitarios y un juego clásico, todo en un mismo disco

Basic para expertos

Los programas Basic también están divididos en tres áreas diferentes: Utilidades, Gráficos y Sonido. Resulta curioso, pero todos estos programas Basic están escritos sobre el antiguo intérprete Basic del Amiga 1000, de modo que a veces hay problemas para que funcionen correctamente con las nuevas versiones (1.2 y 1.3). Entre las utilidades se encuentran **POKES** y **RTEST,** rutinas que enseñan el manejo de los ports de entrada/salida mediante instrucciones POKE, algo muy interesante para los programadores de Amiga. Además hay otros dos programas donde se pueden apreciar estas utilidades: **SERPORT** y **TERMX2,** un pequeño emulador de terminal con protocolo de transferencia de datos en serie.

En el aspecto de sonido, hay tres demostraciones: **Fillibuster, tom** y **tom2.** La primera habla con la síntesis de voz del Amiga (muy simple) y las otras dos son demostraciones digitalizadas de sonido de tambores, muy curiosas. Finalmente en cuanto a los gráficos, hay una larga serie de pequeños programas ilustrativos sobre las posibilidades de animación y dibujo con el Amiga en Basic, un poco limitadas, naturalmente.

Los programas en C son una buena fuente de aprendizaje para todos los programadores que quieren «destripar» los listados de programadores veteranos. En este disco está el listado fuente de una de las primeras versiones de **MORE,** el conocido utilitario que permite visualizar ficheros de texto de múltiples maneras. También hay listados fuentes que enseñan cómo aplicar el C al manejo de los **Gadgets** del sistema operativo del Amiga, un aspecto verdaderamente muy interesante. Y también hay listados sobre la creación de sonidos directamente desde C.

CLI: nuevas herramientas

Las herramientas para el CLI son bastante típicas, y se suman a las muchas existentes en el campo del dominio público. Por ejemplo, **Timer** es una especie de temporizador que permite realizar esperas, **Grep** es un nuevo comando del tipo «buscadatos-en un-fichero» y **LS** un listador de ficheros (como DIR) que presenta una salida muy similar a la del sistema operativo UNIX.

Otros dos pequeños programas completan la colección: **WildExp** y **UtilSkel.** Ambos permiten emplear nuevos comandos «wildcard» (comodines) en los nombres de fichero, además de los ya incluidos en el AmigaDOS. Puede ser práctico para aquellos que estén acostumbrados a otro tipo de comodines, como los de MS-DOS.

Hack: Aventuras en las Mazmorras

Lo que realmente se lleva la palma en el disco BCS-AMIGA 4 es el juego **Hack.** Se trata de una versión para Amiga del clásico **Rogue** de los sistemas Unix. La historia de este juego es larga, ya que comenzó como un pequeño entretenimiento y terminó como un auténtico juego de acción y estrategia con muchos niveles, objetos, pasadizos, frases, acciones y muchas cosas más. Rogue se ha convertido en un auténtico clásico de los juegos de dungeons & dragons (dragones y mazmorras) y hay gente que lo juega hasta a través de modem.

El juego transcurre en una enorme cueva con 20 niveles diferentes. Cada nivel es distinto del anterior, y todos los niveles

```
float
degrad(degrees)
float degrees;
{
        float pi;

        pi = 335.0 / 113.0;   /* chinese approximation */
        pi *= degrees;
        return(pi/180.0);
}

main(argc , argv)
char *argv[];
{
        int i;
        struct IntuiMessage *message;
        if (argc) is_cli = 1;

#ifdef  MY_DEBUG
        if ((dfp = fopen("debug.file" , "w")) == NULL) {
            if (is_cli) printf("can't open debugging file\n");
            exit(1);
        }
        fprintf(dfp,"debugging information\n");
        fflush(dfp);
#endif
```

cambian en cada partida. En cada pantalla hay siempre varias habitaciones, que están conectadas mediante pasillos o pasadizos. El objetivo del juego es recorrer todos estos niveles recogiendo tesoros, matando monstruos y encontrando diversos objetos que ayudan a proseguir en la aventura.

Lo más sorprendente del juego son los gráficos... ¡no hay! al ser éste un juego pensado como «pura fantasía» y preparado para correr bajo cualquier ordenador, los personajes y pasadizos han sido sustituidos por letras, símbolos y caracteres gráficos. Así, el protagonista es una letra «@», los pasadizos, «#», el perro que te acompaña, una «d», el tesoro, «$», y así sucesivamente. ¡Hay que tener mucha imaginación para jugar!

Los movimientos y acciones se realizan desde el teclado: con las teclas del cursor se mueve el personaje, y cada letra realiza una acción, por ejemplo «f» para buscar objetos o «w» para seleccionar un arma. Algunas acciones van acompañadas de mensajes explicativos o frases de alguno de los personajes. En algunos de los niveles hay tiendas, donde se pueden comprar armas y pócimas, así como otros objetos variados.

Junto al protagonista se mueve un pequeño perro acompañante, que permite despreocuparse un poco de la retaguardia, aunque hay que tener siempre cuidado para no atacarle. Los enfrentamientos con los monstruos se realizan siempre en casillas en las que coinciden el protagonista y algún monstruo, y se saldan a favor de uno u otro dependiendo de la fuerza, la armadura y las armas empleadas. El juego en su conjunto se compone de todas las aventuras por los diversos niveles, cientos de objetos mágicos y gran variedad de monstruos, cada uno con su propia personalidad.

Sin duda **Hack** es un juego realmente entretenido especialmente para los jugadores con una gran imaginación, que sepan transportarse a un mundo fantástico lleno de letras y signos ortográficos en vez de gráficos en 4.096 colores. Todo un clásico muy recomendable.

¡VIRUS FUERA!

De todos son conocidos los problemas de tener un virus en los discos que habitualmente utilizas con tu ordenador: ficheros que desaparecen, discos corrompidos, pantallas «juguetonas», cuelgues inesperados... Afortunadamente, los veteranos programadores de Amiga han creado un gran número de vacunas antivirus que permiten «proteger» tanto los discos como el ordenador de posibles infecciones.

Estos pequeños «mata-virus» se suelen encontrar en algunos discos que ruedan por los círculos informáticos, junto con programas utilitarios y copiadores (recomendados: MipMap y The Force). Algunos de los mejores son los siguientes:

Casi todos los usuarios de Amiga se han encontrado con el desagradable problema de los molestos virus.

VirusX. Al arrancarlo aparece una pequeña ventana en el Workbench, y cada disco que se inserta, en cualquiera de las unidades (DF0:, DF1:, etcétera) es chequeado. Si contiene un virus, aparece un requester que permite eliminarlo. Reconoce virus como el SCA, Byte Bandit y otros, además de boot-sectors alterados. Es de los mejores, cómodos y rápidos. Se puede instalar en el s/startup-sequence para mayor seguridad al trabajar con el workbench.

V.I.P. Es un programa comercial (no DP) de Discovery Software que también detecta y mata virus que haya en los discos. Con una bonita presentación.

Además de éstos hay toda una serie de antiguos protectores, como son **Virus Killer** (con música de Queen), **Virus Hunter, Virus Killer FvL, SCA Protector** (muy espectacular y también con música), **Odin-viruskiller** y otra larga serie. Todos ellos son del tipo «inserta el disco a comprobar», y señalan si tienen o no virus.

Entre los programas de tipo «preventivo» hay varios que quedan instalados en el ordenador y lo protegen tras los resets o al cargar cada disco. El más conocido y mejor de ellos es **The Guardian.** Tras instalarse, chequea todos los discos de arranque (con boot-sector) y si encuentra algún problema, algún vector modificado o cualquier otra cosa extraña lo notifica y permite corregirlo o instalar el disco para matar el virus. Son similares**Z Virus Protector** y **Tristar.** Todos ellos tienen la característica de emitir una pequeña «musiquilla» al arrancar tras el reset, como en el Amiga 1000, y algunos hacen rayas de colores para mostrar su presencia.

Finalmente, recordar que un buen detector de virus es cualquier programa que permita visualizar el sector de arranque del disco, como por ejemplo **Disk-Wik.** Y para matarlos, un buen **INSTALL DRIVE DF0:** desde el CLI suele ser una buena solución. Y recuerda comprobar todos los discos que te presten antes de utilizarlos. ■

Los programas comentados en esta sección son Dominio Público. Pueden conseguirse copias de ellos a través de amigos, grupos de usuarios, clubs, BBSs o compañías especializadas.



EXPANSION DEL BUFFER DE TECLADO

El buffer de teclado que posee el C-64 es bastante reducido. Si se utilizan rutinas de teclado dinámico, este problema representa una gran limitación. Los diez caracteres de extensión que posee el mencionado buffer se quedan cortos en la mayoría de los casos prácticos. Esta rutina es una utilidad en modo interrupción que permite expandir la capacidad del buffer de teclado hasta un total de 255 caracteres. Por supuesto, lo primero que hace falta es relocalizar la posición del buffer dentro de la memoria, pero de eso también se encarga el programa.

Después de teclear y grabar este programa, ejecútalo para ver su funcionamiento. Se puede incorporar a cualquiera de tus propias rutinas. Se deben desconectar los cartuchos para su correcta utilización.

```
PROGRAMA: EXPANSION                    LISTADO 1

0 REM EXPANSOR DEL BUFFER DE TECLAD .208
0 64
1 :                                  .233
2 REM COMMODORE WORLD 61             .216
3 :                                  .235
5 REM RICHARD PENN                   .119
7 :                                  .239
10 A=88:B=255:C=56334:D=65535:POKE1 .166
,55:POKEC,PEEK(C)AND(B-1)
20 POKEA+7,0:POKEA+8,160:POKEA+2,B: .58
POKEA+3,B:POKEA,B:POKEA+1,B:SYS4191
9
30 POKED,PEEK(D):POKEC,PEEK(C)OR1:P .4
OKEC-9,56:POKE1,53
40 POKE58805,0:POKE58806,192:POKE58 .16
810,1:POKE58811,192
50 POKE58813,0:POKE58814,192        .28
60 POKE60221,0:POKE60222,192:POKE64 .98
9,255
65 :                                 .41
70 REM DEMOSTRACION                  .76
75 :                                 .51
80 PRINTCHR$(147)"DEMOSTRACION DEL  .112
BUFFER AMPLIADO"
90 FORT=49152TO49407:POKET,65:NEXT: .128
POKE198,255
```

CARGADOR DE DISCO CON FLASH

Mediante la sencilla rutina que presentamos a continuación se pueden animar las cargas de programas de disco, tanto desde modo directo como desde tus propios programas.

Cuando el C-64 comienza la carga de un programa de disco, el borde de la pantalla comienza a parpadear variando los colores y creando el famoso efecto de las rayas. Este efecto continúa hasta que la carga del disco termina. Por lo que nosotros hemos comprobado, la carga no afecta a los programas, aunque puede haber algún caso. Por otra parte, se debe desconectar el Final Cartridge II ya que este último tiene otras rutinas de carga.

Pulsando RUN/STOP y RESTORE se desconecta la rutina, que se puede reactivar mediante POKE 1,53.

```
PROGRAMA: DISCO FLASH                  LISTADO 2

0 REM FLASH DISK LOADS               .70
1 :                                  .233
2 REM COMMODORE WORLD 61             .216
3 :                                  .235
5 REM RICHARD PENN                   .119
7 :                                  .239
10 A=88:B=255:C=56334:D=65535:POKE1 .166
,55:POKEC,PEEK(C)AND(B-1)
20 POKEA+7,0:POKEA+8,160:POKEA+2,B: .58
POKEA+3,B:POKEA,B:POKEA+1,B:SYS4191
9
30 POKED,PEEK(D):POKEC,PEEK(C)OR1:P .4
OKEC-9,56:POKE1,53
40 FORT=679TO679+8:READD:POKET,D:NE .184
XT
50 DATA238,32,208,120,169,0,76,22,2 .32
38
60 POKE60947,76:POKE60948,167:POKE6 .146
0949,2
```

OTRO MODO DE HACER INPUT

Para eliminar el signo de interrogación de los comandos INPUT en tus programas, puedes utilizar esta rutina. Es útil tanto en el C-64 como en el C-128. Se abre un canal de teclado mediante OPEN 2,0 y con una rutina «input» desaparecen los signos de interrogación.

El programa realiza la operación con el canal de teclado y se demuestra a sí mismo con una sencilla pregunta.

```
PROGRAMA: OTRO INPUT            LISTADO 3

0 REM OTRA FORMA DE INPUT               .162
1 :                                     .233
2 REM COMMODORE WORLD 61                .216
3 :                                     .235
5 REM CHAD HELFENBERGER                 .37
7 :                                     .239
10 OPEN2,0                              .226
20 PRINT"CUANTO ES 5 POR 5"             .232
30 INPUT#2,T                            .14
40 IFT<>25THENPRINT"NO, "T" ES INCO     .134
RRECTO":CLOSE2:END
50 PRINT"[CRSRD]HAS ACERTADO, "T" E     .36
S CORRECTO"
60 CLOSE2                               .80
```

METEOR DODGE 64

Este jueguecito escrito en Basic es muy sencillo, pero sirve para entretenerse, aunque sea un ratito corto. Mediante las teclas «,» (coma) y «.» (punto), se mueve el diamante azul de la parte superior de la pantalla. El juego consiste en esquivar los meteoros (asteriscos) que van llenando la pantalla.

Las variables utilizadas son: SC para la puntuación, S para la posición del diamante, C para el color del rombo, L para el nivel de juego y X para el número de meteoros por línea.

```
PROGRAMA: METEOR DODGE          LISTADO 4

0 REM METEOR DODGE                      .214
1 :                                     .233
2 REM COMMODORE WORLD 61                .216
3 :                                     .235
5 REM GARRET WILSON                     .79
7 :                                     .239
10 PRINT"[CLR][BLU]":POKE53280,0:PO .82
KE53281,0:SC=0:C=55316:DIMR(40):L=1
:X=10
20 S=1044:PRINT"[8CRSRD]"TAB(16)"LE .94
VEL 1":FORA=55316TO56295:POKEA,1:NE
XT
30 FORN=1TO2000:NEXT:PRINT"[CLR][27 .204
CRSRD][WHT][CRSRD]":FORT=1TO75
40 POKES,90:POKEC,6:SC=SC+1:FORG=1T .150
OX:R(G)=(1984+(INT(RND(0)*40))):GOS
UB120
50 POKER(G),42:NEXT:PRINT"[CRSRD]"; .0
60 IFPEEK(S)=42THENPOKEC,6:GOTO90   .36
70 NEXT:L=L+1:PRINT"[CLR][BLU][12CR .126
SRD]"TAB(16)"LEVEL"L:X=X+2:IFX>39TH
ENX=40
80 GOTO30                               .20
90 POKES,102:FORN=1TO2000:NEXT:PRIN .40
T"END OF GAME":PRINT"SCORE= "SC
100 POKE198,0:INPUT"PLAY AGAIN (S/N .16
)";PA$:IFPA$="S"THENRUN
110 END                                 .112
120 GETM$:P=PEEK(56320)                 .18
130 IF(M$=","ORP=123)ANDS=1024THENS .66
=1063:C=55335:M$="":POKE1024,32:POK
E55296,1
140 IFM$=","ORP=123THENS=S-1:C=C-1: .210
POKES+1,32:POKEC+1,1
150 IF(M$="."ORP=119)ANDS=1063THENS .24
=1024:C=55296:M$="":POKE1063,32:POK
E55335,1
160 IFM$="."ORP=119THENS=S+1:C=C+1: .192
POKES-1,32:POKEC-1,1
170 IFPEEK(S)=42THENPOKEC,6:GOTO90  .146
180 POKES,90:POKEC,6:RETURN         .182
```

AUTOARRANQUE DE LA UNIDAD 1581

Los usuarios del Commodore 128 que disponen además de launidad de disco 1581, tienen diversos problemas para suutilización. Especialmente complicado es el autoarranque deficheros grabados enlos discos de 3,5 pulgadas. Además, losusuarios del modelo 128D, no disponen de conmutadores en la 1571interna para cambiar el número de periférico. Además, se debedesactivar la 1571 para que la 1581 autoarranque.

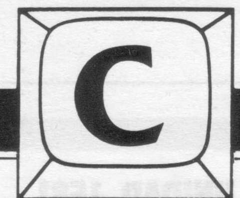
Existe un truco para hacer un fichero de 1581 autoarrancable:

10 RUN«FICHERO 1581»,U9

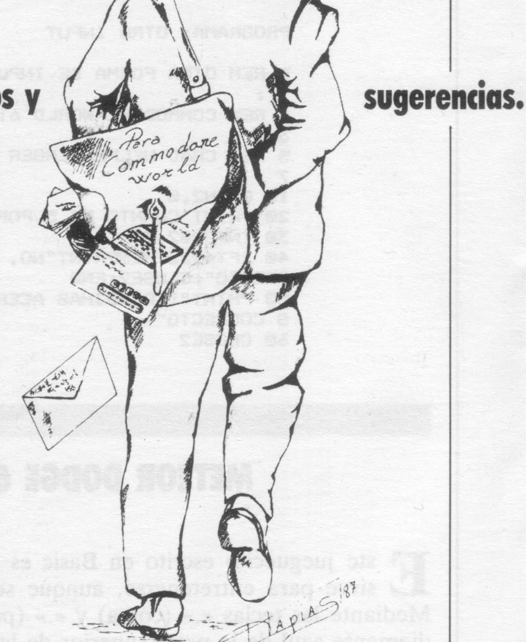
Grabando ese fichero en la 1571 y utilizando el programa«Autoboot Maker», se puede tener el fichero que se deseaautoarrancar en la 1581 y este fichero «arrancador» en el discode la 1571. También se pueden tener autoarranques en ambasunidades de disco mediante una simple línea:

10 BOOTU9

El método es sencillo y muy útil. También los usuarios de 1581 tienen sus magias.



ARTAS DEL LECTOR



Esta sección es un buzón abierto a vuestras consultas, comentarios y sugerencias.

AMIGA EN CASTELLANO

1. ¿Se le puede agregar al Amiga un buen procesador de textos que escriba con los caracteres castellanos? Es decir, acentos, diéresis, interrogación y admiración invertidas, etcétera.

2. ¿Viene el Amiga 500 preparado para escribir en castellano?

3. Para hacer lo indicado en las anteriores preguntas, ¿es necesario alguna impresora especial o por el contrario sirve cualquier impresora compatible como la BROTHER M-1109, que es la que tengo? Debo decirles que dicha impresora tiene dos modos, compatible Epson y compatible IBM. También tiene dos entradas: Centronics y RS-232 C.

Rafael Peillón Jubillar
Zaragoza

Casi todos los procesadores de textos para Amiga (y otros ordenadores) permiten la utilización directa de caracteres internacionales, como son los acentos y los signos ortográficos especiales del castellano. Por otro lado, el Amiga 500 se vende con el teclado castellano, como lo imponen las leyes de homologación en nuestro país.

En cuanto al procesador de textos, el más recomendable es sin duda *WordPerfect*. Las múltiples definiciones para impresora que incluye te permitirán poder trabajar en castellano perfectamente. Como siempre, cualquier otro tipo de problemas con la impresora (drivers, selección interna, etcétera.) deberás solucionarla por tu cuenta con mucho cuidado y algunos conocimientos de programación.

IMAGEN, SONIDO Y PROGRAMAS

1. Tengo un monitor SONY KV-1442 con pantalla negra y euroconector. ¿Este monitor es compatible con el Amiga? Si no lo es directamente, ¿pueden decirme algún interface para esto que no sea vergonzosamente caro?

2. ¿Está disponible en España el programa KWIK-SPEAK 1 para Amiga?

3. ¿Dónde se conecta el cassette al trabajar con el emulador Go 64?

4. ¿Conocen algún programa para Amiga para llevar la contabilidad de una farmacia como existe para PC's?

Hilario García Ostos
Córdoba

1. El Sony Trinitron con euroconector es compatible con el Amiga en modo RGB (contrariamente a lo que nosotros mismos indicamos hace algunos números en esta misma sección). Para ello hace falta un cable apropiado que puedes construirte tú mismo (ver Commodore World 57) o comprarlo en alguna tienda especializada en productos para Amiga.

2. Según nuestra guía de software (ver Amiga World 2), ese programa no lo comercializa todavía nadie en nuestro país, aunque algunos importadores, como *ABC Analog*, pueden importarlo para ti directamente de Estados Unidos.

3. Ni el *Go 64!* ni el *64 Emulator* permiten conectar el cassette al Amiga. Por lo tanto si a estas alturas todavía no dispones de unidad de discos (¿todavía no?), lo mejor será que localices a alguien que te pueda transferir los ficheros a disco. Las unidades de disco se conectan a través de un interface especial. Por otro lado, siempre existe la posibilidad de pasar los ficheros por cable o modem con un programa de comunicaciones apropiado.

4. Desgraciadamente, el mercado para Amiga todavía no es suficientemente amplio como para que aplicaciones de tipo «vertical» como la que nos comentas hayan hecho su aparición todavía. Lo más parecido es el *ContAmiga* de Barnacomputer.

FONTS Y CAPTURA DE IMAGENES

1. ¿Cómo debo hacer para asignar la carga de los fonts en un disco que no sea el del sistema? Por favor explíquenmelo detalladamente.

2. ¿Existe algún programa que consiga capturar imágenes de programas como Operation Wolf e imprimirlas por impresora?

3. ¿Me podrían decir el título de algún libro que trate del Assembler o Código Máquina del 68000?

David Parcerisa Carenas
Barcelona

1. Lo único que hay que hacer para cambiar la asignación de los fonts, o de cualquier otro «elemento» asignable, es teclear desde el CLI la instrucción: ASSIGN <asignación> <dónde–se–asigna>

En tu caso, si quisieras asignar los fonts, cuyo identificador es FONTS: a un disco llamado «MiDisco:», la instrucción sería: *ASSIGN FONTS: MiDisco:Fonts*. Si el subdirectorio tuviera otro nombre, por ejemplo, Fuentes/Grandes, la instrucción sería: ASSIGN FONTS: MiDisco:Fuentes/Grandes.

También puedes cambiar la asignación arrancando con el disco en el que tienes los fonts. Por ejemplo, para utilizar los fonts de TV*TEXT con Deluxe Paint II, bastaría con arrancar con TV*TEXT y, desde la pantalla del Workbench, cambiar de disco y arrancar Deluxe Paint II. El Amiga se encarga más adelante de pedir los cambios de disco.

2. Hay un gran número de programas muy antiguos, como *Grabbit!* o en el dominio público *ScreenX*.

3. El libro que buscas es 68000, 68010, 68020 Arquitectura y Programación en Ensamblador, de Anaya.

ESCUELA DE PROGRAMADORES

Hola, amigos de Commodore. Me gustaría saber programar juegos comerciales en Código Máquina; así que me gustaría encontrar alguna academia

donde enseñen esto, si hay algún libro o publicación que hablara sobre ello o si podrían ustedes decirme si se podría hacer una sección en la revista dedicada a este aspecto de la programación.

Domino el lenguaje máquina, pero me faltan conocimientos tales como:

- Utilización de interrupciones en los juegos.
- Técnicas de mapeado.

Y algunas otras más.

Sigo el Commodore World desde el número 1; por las secciones dedicadas a C.M. he aprendido este lenguaje, pero nunca he visto un artículo que me solucionara realmente estas dudas.

Tomás Escartín Gutiérrez
Madrid

No resulta difícil comprender que no exista ninguna academia de este tipo, puesto que la gran mayoría de los programadores profesionales tanto de juegos como de aplicaciones son autodidactas al máximo. La mejor enseñanza es la que se obtiene tras muchas horas delante del teclado, leyendo revistas y libros especializados, tanto en castellano como en inglés y «destripando» los programas de los demás. No existen libros específicos que enseñen la programación de juegos comerciales en código máquina. Hay libros de juegos pero son sólo orientativos y bastante anticuados, con listados incluidos de juegos sencillos en Basic.

En la revista llevamos un par de meses publicando las rutinas de un futuro juego comercial, *Sabrina,* adaptación del mismo juego de Spectrum, en donde el programador explica las técnicas utilizadas para la creación de gráficos, movimientos y, en los próximos números, mapeado y otros aspectos del juego completo.

Te recomendamos que releeas los artículos con rutinas especiales escritos por José Manuel Fluxá: *La Bolsa* (46) *Efectos Especiales* (49,52), *A toda pantalla* (48) y *Stars* (50), donde se aprovechan a fondo las interrupciones.

DEL C-64 AL AMIGA

Soy un asiduo lector de su revista pues soy propietario de un C-64. Hace un par de semanas he adquirido un Amiga-500 con la ampliación de memoria A-501. Entra dentro de mis pretensiones la adquisición de un monitor (en color por supuesto), una segunda unidad de discos y, cuando la economía lo permita, una impresora en color.

Mi primera confusión (que agradecería me comentaran), surge a raíz de una carta del lector aparecida en el n.° 56 de la revista. En ella se habla de los monitores y sus vibraciones. Para mí, de entrada, es imprescindible que el monitor sea estéreo (es decir, el 1084S), pero ustedes dicen que la pantalla no es mate y consiguientemente, vibrará. Esto me asusta. ¿Cuánto vibra? ¿Es un ligero parpadeo que moleste a las mentes sibaritas, o es una cosa que un usuario sin pretensiones como es mi caso, pueda aguantar? He observado que en alta resolución, y con un pequeño televisor, la pantalla vibraba efectivamente arriba y abajo siendo del todo insoportable. En un monitor como el 1084S ¿me va a pasar lo mismo y en la misma medida? ¿Debo sacrificar el estéreo y adquirir el 1084 en pantalla mate? En este caso ¿existe alguna forma rápida, cómoda y barata para un «no manitas en electrónica» de acoplar un par de altavoces para la salida estéreo?

Por otro lado, en la revista número 52, también en la sección de cartas del lector, ponéis así entre paréntesis, que la unidad de discos del C-64, la 1541 puede usarse con el Amiga. Este tema me deja harto sorprendido pues ni tan siquiera podía a mí ocurrírseme pensar que ello fuese posible. Lo considero algo muy importante, pues permitiría un acceso cómodo al formato de 5 1/4″ propio de los PC's, pues no me parece aconsejable comprar una segunda unidad que no sea de 3 1/2″, pero claro, podría guardarme la 1541 como tercera unidad y sería fabuloso.

Por la importancia del tema, deduzco que ya lo habréis tratado en la revista, pero aunque tengo prácticamente todas, no encuentro nada al respecto de cómo «enganchar» la 1541 al Amiga. Agradecería pues que me explicaseis cómo hacerlo.

Por último, en el número 57 aparece un interesante artículo sobre la MPS 1230 y MPS 1250. Tengo una MPS 1200 (o sea, una Citizen 120D cambiada de nombre), con el cartucho interface correspondiente para el C-64. Lamentablemente, no encuentro referencia a ella en el artículo y me gustaría saber si estoy muy equivocado al pensar que con un cartucho Centronics y el cable correspondiente puedo conectarla al Amiga. Si ello no me supone mucho dinero, me gustaría hacerlo, aunque sea temporalmente hasta poder comprar una impresora mejor para el A500.

Un detalle más (con perdón). ¿Qué diferencias prácticas son de esperar entre los ordenadores donde aparece la carátula de V1.2 y de V1.3? Gracias.

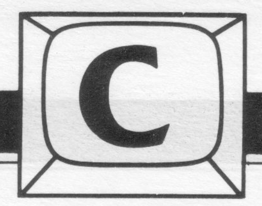
Francisco López Baldovin
Córdoba

La mejor forma de oír el Amiga en estéreo es conectándole a una cadena o amplificador con grandes altavoces. Esto se logra con un sencillo cable entre la salida de sonido del Amiga y la entrada auxiliar de la cadena. Los resultados son asombrosos. Este mismo efecto puede conseguirse con cualquier pequeño amplificador de los que venden en tiendas de electrónica y un sencillo par de altavoces. El 1084S es bastante peor monitor que el 1081 en cuanto a imagen, y las vibraciones en el modo interlazado son similares e inevitables. Se pueden aguantar las vibraciones durante algún tiempo, pero es muy cansado para la vista y poco recomendable. El 1084 mate tampoco es la solución ideal. La solución ideal se llama *Flicker-Fixer* y tiene un «módico» precio de casi 100.000 pesetas. Aquí no hay soluciones caseras.

Para adaptar la 1541 al Amiga hace falta un conector y software especial, que por desgracia todavía no hemos podido probar (¡lo sentimos mucho!). Lo que es seguro es que ese interface no permite utilizar el formato PC, sino únicamente el formato Commodore, es decir… 170K y gracias.

Con un interface Centronics y la impresora no hay ningún problema a la hora de imprimir, excepto los clásicos de los «malditos» drivers.

Entre las ventajas prácticas están: Con la ROM 1.3, autoarranque de algunos discos duros y sistema operativo más depurado (menos «gurus»). El Workbench 1.3 mejora ostensiblemente en velocidad (FastFonts), edición en el CLI, nuevos utilitarios, comandos mejorados y ampliados y más posibilidades de expansión y control sobre el ordenador. ■





AEGIS AUDIOMASTER II

Ordenador: Amiga
Fabricante: Aegis Software
Distribuidor:
Pixel Soft
Plaza Isabel la Católica, 6
34005 Palencia
Tf: (988) 75 11 80

Las nuevas versiones de los programas clásicos van apareciendo en el mercado poco a poco, como es el caso de AudioMaster II. Este conocido programa para la edición y manipulación de sonidos digitalizados se ha visto recientemente actualizado y mejorado en algunos aspectos importantes, como es la edición en estéreo. Respecto al banco de pruebas de la versión 1.0, puedes consultar el número 51 de Commodore World.

La música digitalizada inunda literalmente todos los programas del Amiga, desde los juegos hasta las presentaciones y las demostraciones de productos. Es relativamente fácil para un usuario con un equipo sencillo conseguir buenos resultados sin necesidad de recurrir a complejos programas. AudioMaster II es un buen ejemplo de sencillez y a la vez potencia para los aficionados al sonido.

Hardware necesario

Lo primero que es necesario para trabajar con AudioMaster II es un aparato digitalizador. Sin él, todo lo que se podrá hacer es manipular sonidos ya existentes, como los de demostración, juegos comerciales y similares. Esto es lo primero que ha de tenerse en cuenta a la hora de comprar este paquete.

Actualmente es posible encontrar en el mercado nacional todo tipo de aparatos digitalizadores de muy buena calidad a un precio razonable. Entre ellos se encuentran **Perfect Sound, TH-Sound,** y otros similares. Todos ellos permiten la conexión a un equipo de sonido o a un micrófono, desde donde se obtiene la fuente de sonido a digitalizar. Si el aparato es bueno, la digitalización carecerá prácticamente de ruido, y el resultado final será perfecto.

Algunas innovaciones importantes

Dentro del propio programa, nada más arrancar el disco con AudioMaster II se observan algunas mejoras externas en el aspecto de la presentación, como son la pantalla PAL (por fin) y nuevas opciones los menús. La más importante novedad de todas es la incorporación del sonido en **estéreo.** Esto permite manipular sonidos digitalizados en dos bandas distintas, de modo que el sonido resultante se oye por el canal de la izquierda y el de la derecha de forma distinta, proporcionando efectos agradables al oído. El sonido estéreo es una de las capacidades del Amiga que menos han sido aprovechadas, tal vez por falta de medios. La mayoría de los programas se limitan a hacer unas «simulaciones» de estéreo, tocando la música por los dos canales, sin que sea un estéreo real.

La edición en estéreo es muy similar a la normal, sólo que la pantalla se divide horizontalmente en dos planos que simbolizan cada forma de onda. Todas las operaciones de edición se realizan siempre en zonas equivalentes para las dos vías.

Aegis ha sabido respetar la configuración y el aspecto externo de AudioMaster, de modo que todas las nuevas opciones se han añadido sobre las antiguas, sin modificar considerablemente el funcionamiento general del paquete. Los comandos con abreviaturas (shortcuts) desde el teclado también son los mismos, y respetan la filosofía general del Amiga según lo establecido.

Además del programa, es necesario conseguir un buen aparato digitalizador, preferiblemente en estéreo.

Digitalización de sonidos

El proceso general que se emplea para la digitalización de sonidos es bien simple: en primer lugar se prepara la fuente de sonido y el equipo a digitalizar. Como ya se ha comentado, puede utilizarse cualquiera de los muchos digitalizadores existentes en el mercado. AudioMaster II trabaja tanto con los que se conectan en el port de impresora como con los que se conectan en el port de joystick.

La calidad de la digitalización depende en gran medida del aparato digitalizador empleado, y también de saber utilizar el software correctamente. Si la digitalización se realiza a mucha velocidad, la calidad será mayor, y menor cuanto más baja. La desventaja que supone la mayor velocidad de muestreo es el gran consumo de memoria: un sonido de unos 6 u 8 segundos de duración puede necesitar 100 o más Kbytes de memoria. Para un perfecto aprovechamiento del programa se recomienda 1 Mbyte de memoria (quedan unos 680K libres) aunque también se puede trabajar con un Amiga 500 sin ampliación.

Los muestreos normales (samplings) se hacen a una velocidad de unos 8.363 sps (samplings per second, muestreos por segundo). Se puede aumentar hasta más de 20.000 por segundo, pero no es muy recomendable. Hay que tener en cuenta además que esta velocidad se reduce a la mitad si el muestreo se está efectuando en estéreo.

Nuevas funciones

La pantalla principal de AudioMas-

ter II se compone, al igual que la de la antigua versión, de un menú en la parte superior, la representación gráfica de la onda en la parte central y unos controles para la ejecución del sonido en la parte inferior. Estos gadgets permiten oír el sonido digitalizado, así como marcar una zona y ampliarla (zoom), cambiar el volumen y buscar algunas zonas específicas (ceros) en el gráfico del sonido.

Las funciones básicas de AudioMaster II se dividen en varios menús, cada uno de ellos orientado hacia un campo determinado. Como es habitual, el primero de ellos es Project, desde donde es posible cargar y grabar ficheros de sonido (formato IFF), recurrir al **sampler** (digitalizador) para realizar el muestreo de sonido o salir del programa. Aquí se han incluido dos nuevas opciones: **MODE,** que permite cambiar de modo (modo o estéreo) y **PRINT** que permite imprimir la gráfica del sonido digitalizado.

El antiguo menú de edición ha sido ampliado y dividido en dos nuevos menús, para acoger todas las opciones existentes. Además de las ya habituales para cortar, copiar, borrar y pegar «trozos» de sonido digitalizado, lo que permite un control casi total sobre el sonido. Dos nuevas opciones, Invert y Swap Channel permiten invertir los valores altos por los bajos e intercambiar los datos de cada canal, respectivamente. Como siempre, las funciones para borrar y «dibujar a mano» la forma de onda permiten manipular el sonido directamente.

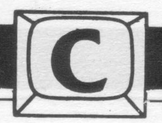
Todas las funciones de edición Audiomaster II son sencillas y muy completas.

Otras nuevas funciones de edición son también **Replicate Loop,** que permite obtener varias copias de una parte del sonido, **Add Workspace,** para insertar una zona en blanco en cualquier punto y **Swap Buffer & Main,** muy interesante, para intercambiar el sonido que se está editando y el que se encuentra en el buffer de cortar/copiar. En conjunto, todas las funciones de edición son muy útiles y rápidas.

El menú de efectos especiales apenas ha sufrido modificaciones. Permite crear efectos de eco, ejecución hacia atrás (verdaderamente curiosa), cambiar el volumen, mezclar los sonidos de la pantalla y el buffer, utilizar un filtro pasa-bajos para resaltar los sonidos de baja frecuencia o cambiar el volumen. La única novedead es **Real Time Echo,** que permite obtener efectos de eco y voces en el tiempo real.

Finalmente, en los menús de **User Options** e **HiFi** también hay algunas novedades, como por ejemplo la definición de los límites de los bucles, un pre-muestreo, filtros de audio y la posibilidad de modificar otros valores, como son los colores de la pantalla y la configuración del sampler.

Algunas de las antiguas funciones han sido modificadas para permitir un mejor aprovechamiento del programa. Así, por ejemplo, la búsqueda en



OMENTARIOS COMMODORE

RAM de sonidos ya existentes (RAM Scan) ha sido dividida por zonas de memoria, lo que permite a los usuarios con ampliación de memoria sectorizar la búsqueda. Esta opción es muy interesante pues permite «capturar» sonidos de programas como los juegos comerciales.

Instrumentos personalizados

Una interesante aplicación de AudioMaster II es la creación de cortes digitalizados a partir de instrumentos de música que se encuentren grabados en una cinta, compact disc o cualquier otro tipo de soporte magnético. El proceso es bien simple: con sumo cuidado se elige el sonido del instrumento, en un fragmento de música en el que no esté mezclado con ningún otro instrumento. A continuación, se pasa por el digitalizador y por el programa AudioMaster II, utilizando el editor para dejar únicamente la porción correspondiente al sonido, sin espacios en blanco.

El siguiente paso es buscar, si es necesario, el «tono» correcto antes de grabar el instrumento a disco, lo que se consigue modificando la onda (Tune Waveform). Una vez grabada a disco con la extensión correcta en el nombre, puede ser cargada desde programas como Aegis Sonix, desde donde se aplicará para formar la escala completa en el teclado.

Existe la posibilidad de digitalizar el sonido de un instrumento y emplearlo en programas de música.

La posibilidad de crear estos instrumentos «a medida» se hace patente desde el momento en que el usuario de los programas de música se pecata de que la cantidad de instrumentos es limitada. Aunque la mayoría de los paquetes musicales incluyen discos es que a la larga siempre se echa en falta la «personalización» de la música.

El próximo número de Amiga World (3) contará con un artículo donde se explican esta y otras técnicas similares. El mundo de la música y los instrumentos a partir de sonidos digitalizados está a tu alcance.

Aplicaciones

AudioMaster II puede considerarse un programa muy necesario en la mesa de cualquier aficionado al mundo de la música y el sonido. Junto con un buen digitalizador de sonido es una herramienta ideal para crear sonido, «cortes» y todo tipo de efectos especiales que pueden aplicarse a presentaciones, slide-shows, juegos, programas en Basic o C, y demás. Esta nueva versión es rápida, precisa, completa y muy recomendable.

De todas formas, como siempre recomendamos, lo mejor es hacer algunas pruebas antes de comprar, es decir, coger el digitalizador e irse a la tienda para poder probar el programa con la misma configuración con la que se va a utilizar posteriormente. De este modo se evitan muchas sorpresas desagradable. Pero en conjunto, Audiomaster II es un producto necesario para todo aquel que emplee un digitalizador. ■

METEDURAS DE PATA

En el capítulo 2 del **Cursillo de Lenguaje Ensamblador** para Amiga (Commodore World 60), no se hace referencia al listado 4 en ningún punto del artículo. En realidad estos tres pequeños listados corresponden a los tres ejemplos del final de la página 45: Cómo situar el cursor del fichero al principio, al final y en una posición determinada del fichero.

En el programa **LOTO** de la revista Amiga World 1 han aparecido dos nuevos pequeños bugs referentes a los filtros, que pueden corregirse como se indica a continuación:

En las líneas de «Cálculo para los filtros» (página 54) hay que modificar las líneas:

```
numpa = numpa + (nu%(j,i) AND 1)
```
por
```
numpa = numpa + ((nu%(j,i)+1) AND 1)
```
y también
```
IF FilF%(7)=1 AND (totcol < Maxcol) THEN GOTO Eliminar
IF FilF%(8)=1 AND (totfil < Maxfil) THEN GOTO Eliminar
```
por
```
IF FilF%(8)=1 AND (totcol < Maxcol) THEN GOTO Eliminar
IF FilF%(7)=1 AND (totfil < Maxfil) THEN GOTO Eliminar
```

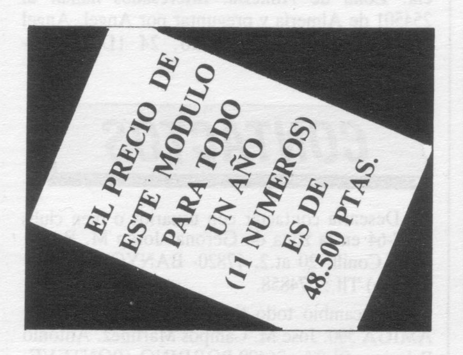
El primer cambio arregla el filtro del «máximo de pares», que en realidad filtraba los números impares. Las dos últimas líneas arreglan los filtros de «máximo de números en filas-columnas» que al encontrarse invertidos filtraban realmente «columnas-filas».

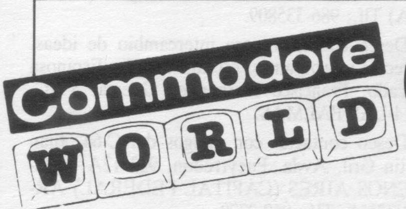
En la **Guía de Software** para Amiga publicada en la revista Amiga World 2, aparecieron unas referencias erróneas en el software de juegos. Los que aparecen con el número 20 como distribuidor en realidad pertenecen al número 19 (sólo en la lista de juegos). Por lo tanto, los programas distribuidos en exclusiva por System-4 son:

African Raiders 01	Emmanuelle	Superman
Asterix en la India	Freedom	The Munsters
Blueberry	La leyenda de Djel	Who Framed Roger Rabbit?
Circus Games	Strip Poker II Plus	
El Libro de la Selva	Summer Olympiad	

La dirección completa de System-4 es: Francisco de Diego, 35, 28040 Madrid. Telf. (91) 450 44 12. Cualquier consulta sobre estos juegos puede dirigirse directamente a ellos.

DIRECTORIO





¿QUIERES COLABORAR CON NOSOTROS?

Si quieres colaborar con nosotros, envíanos tus artículos a la siguiete dirección:

COMMODORE WORLD Colaboraciones.
Rafael Calvo, 18-4.º B. 28010 MADRID.

MARKETCLUB

Servicio gratuito para nuestros lectores *PARTICULARES*. Los anuncios serán publicados durante 1 mes.
Los anuncios gratuitos de Market Club *SOLAMENTE* serán publicados si vienen con nombre, apellidos y dirección completa.

MERCADILLO

- Vendo libros Data Becker para C-64. Colección de 19 títulos, más 10 títulos otras editoriales. Enviaría lista. Escribir a Jose M. Mulero Rimbau. Rovira y Virgili, 10 5º. 43002-TARRAGONA.
- Vendo ordenador C-128, datasette, libros y revistas. Precio a convenir. Enrique Vicente García. Polígono Gornal, 65 4-3. HOSPITALET DE LLOBREGAT-BARNA. Tlf.: 93-3361390.
- Por cambio de equipo, vendo C-64 comprado en octubre del 88, con datassette, manuales y juegos. O lo cambio por impresora en buen estado. Modesto Perez de la Cruz. Margall, 7 5-1. 43480-VILA-SECA. TARRAGONA. Tlf.: 977-390449.
- Compro libros para el C-64. Mandar ofertas a Rafael Santiago Sanchez. Sta. Teresa, 1 1-2. 14800-Priego de Córdoba. Córdoba.
- Vendo C-64 + Cassette + programas + joystick + revistas, todo por 38000 Pts. Contactar con Miguel Carrillo Sanchez. Santander, 29. Algeciras. Cadiz. Tlf.: 655125.
- Ordenador C-128. Datasette. Floppy 1541. Monitor Dinadata, 40 columnas, sonido, fósforo verde. Impresora Seikosa Sp 1000uc. Programas. Juegos en cinta y en disco. Libros DadaBecker. Todo por 100.000 Pts. Ramón García Rojo. Mechía, 63. Jerez de la Frontera. Cadiz. Telf.: 956-346880.
- Vendo impresora EPSON LX-86, con NLQ, alimentador de hojas, unidad de tracción externa, cable y recambios de cinta. Nueva. Oportunidad: 40000 Pts. Llamar por la noche a Rais Soheimann. Telf.: 93-4731817.
- Vendo Amiga 2000 con 9 Mb. de memoria, disco duro de 20 Mb. tarjeta salida de vídeo, todo interno. Alimentador externo para discos, digitalizador de vídeo, digitalizador de sonido y 50 programas originales, con sus respectivos manuales, y el ¡ROM KERNEL! y el ¡INTUITION!. Precio a convenir. José Ramón Fernández Ardura. Sta. Rosalía, 94 2. 08035-BARCELONA. Tlf.: 93-4204758.
- Vendo C-64 NEW, unidad de discos 1541, impresora MPS 801, datasette, juegos, aplicaciones y revistas diversas. Todo por 70000 Pts. Llamar Tlf.: 93-4276514 (Noches). Jorge Calvo. Rda. Guineueta Vella, c-7 3-4. 08033-BARCELONA.
- Vendo C-64 con unidad de disco 1541-D, Final Cartridge II, datasette Commodore, programas en disco con archivadores incluidos y revistas. todo en perfecto estado por 75000 Pts. Juan José del Bosque Villalonga. Po. Valldaura, 192-196 7-1. 08031- BARCELONA. Tlf.: 93-3539739. Llamar por las noches.

CONTACTOS-CLUBS

- Vendo 1250 Pokes para centenares de juegos, más 650 Pokes de programación (para C-64/128), por solo 500 Pts. Mandar a Fco. Exposito. Avda. Carlos V, 24 8B. 28936-MOSTOLES (MADRID).
- Vendo colección de libros Data Becker: Lenguaje Máquina Avanzado, Floppy 1541, Gráficos para C-64, 64 Interno, Pics í Pokes, Juegos y Trucos para el C-64, etc... por 12.000 pts. Juan Jose Gomez. Apostol Santiago, 49. 28017-Madrid. Tlf.:7479600 Ext. 2611.
- Vendo impresora RITEMAN C+ NLQ en perfecto estado, 35000 Pts. Regalo interface Buffer Stack y juegos para el Commodore 64 así como algunos libros de programación para este ordenador. Javier Guerrero Diaz. Emilio Santacana, 5. 11201-ALGECIRAS-CADIZ. Tlf.: 956-630643.
- Vendo libros para el C-64 por cambio de ordenador: 64 Interno, Lenguaje Máquina para Avanzados, etc... También algunos libros de informática y electrónica. David L. García Molina. Antonio López, 51 9D. 28019-MADRID. Tlf.: 4694626.
- Vendo C-128, 30000 Pts. Unidad de disco 1571, 30000Pts. Si compras todo te regalo impresora MPS-801, cassette, joystick, programas de utilidades y juegos. Jose Casais. Ponte de Saa, 74. 27800-VILLALBA-LUGO.
- Vendo C-64, 1571, datasette, monitor (HANTAREX), joystick, revistas, libros Data Becker y discos con programas variados (todos con instrucciones). Precio a convenir. Jim Juncosa Ollé. Marqués de Mulhacén, 2 6I. 08034-BARCELONA. Tlf.: 2039424.
- Vendo C-64, datasette, unidad de discos 1541, impresora Super Riteman F+, Final C. II, joysticks, Discos, Cassettes, funda para impresora, interface copiador. Todo en perfecto estado, por el precio de 80000 Pts. Luis Santapau Salvador. Cartagena, 293 Bis At.1.08025-BARCELONA. Tlf.: 2557291.
- Compact Disc marca Philips con pletina, cassette y radio, acepta compact singles, todo STEREO Y PORTATIL. Lo cambiaría por una unidad de disco para Commodore 64+ Programación. Jose Guinart. Patxillardegui, 10 6b.20014-LOIOLA-SAN SEBASTIAN (GUIPUZCOA). Tlf.: 371313.
- Vendo C-64, unidad de discos 1571, cassette 1531, joystick, cartucho Final C. III, revistas y juegos. Valorado en 140000 Pts. lo vendo por 97000 Pts. David Felipe Villa. Can Valero, 18. 07011-PALMA DE MALLORCA-BALEARES. Tlf.: 971-296418 y 724895.
- Vendo impresora ROBOTRON, para C-64, sin apenas uso, utiliza carretes de máquina de escribir (super económicos), mínimo nivel de ruido, rápida y cómoda de utilizar. Admite papel continuo y hojas sueltas. Todo por 45000 Pts. NEGOCIABLES. También vendo unidad de discos 1571 para C-64/128. Poco uso. Se acompaña de programas. Solo 30.000 Pts. Ramón Fregda Navarro, Pza. la Sol, 9. 25007-LLEIDA
- Vendo revistas Commodore World desde la 1 hasta la actual. 100 Pts. revista. También vendo lote de libros: Lenguaje Máquina para C-64, 64 Trucos y Consejos, Programación avanzada del C-64, Forth para micros y un libro sobre impresoras. Todo el lote de libros (Data Becker) por 6000 Pts. Jose T. Cuello García, Arriba, 9 y 11 5D. 37002-SALAMANCA. Tlf.: 218754.
- Vendo bien cuidados, con sus embalajes y libros, C-128 (28000 Pts.). Regalo: Final C. II y juegos. Unidad de disco 1571 (30000 Pts). Regalo: Loto Super-pro, Contabilidad personal, etc... Riteman C+ (35000 Pts.) con sus libros. Monitor Commodore 1900 (14000 Pts.) con 4 meses, regalo cable 40/80 Columnas. Federico González Zárate. Nueva Torrevieja, 218. TORREVIEJA-ALICANTE. Tlf.: 96-5716418, llamar de 9,30 a 10,30 h. NOCHE.
- Vendo ordenador C-128, datassette, joystick, cintas, libros Data-Becker, revistas y disco CP/M, todo en perfecto estado. Precio a convenir. Miguel Angel Rojo. Torres Quevedo, 9 11C. 39011-SANTANDER. Tlf.: 942-332424. Llamar a partir 21h.
- Vendo impresora RITEMAN C+, más procesador de textos y base de datos. Todo 25000 Pts. Xavier Puigdomenech García. Dos de Maig, 204 5-2. 08013-BARCELONA.

CLUBS

- CLUB USUARIS COMMODORE-TARRAGONA, centro homologado por la Generalitat de Catalunya. Pídenos información de nuestras secciones de: Biblioteca, Programoteca, Cursos, Asesoramiento, Comunicaciones, etc... Seis años al servicio de nuestros socios. Apartado de Correos, 176. 43080-TARRAGONA. Local Social, Fortuny, 4 2-2. 43001-TARRAGONA.
- Estoy interesado en cambiar ideas y formar un Club para el Amiga-500. Interesados escribir a Santiago Ibarra Jiménez. Juan Carlos I, 60 4B. 30800-LORCA. MURCIA. Tlf.: 968-466803.
- Desearía formar un club de usuarios de C-64 de Alzira y alrededores, para intercambio de experiencias e ideas. Bernardo J. March Pérez. Dr. Fleming, 44 2-4. 46600-Alzira. Valencia.
- PLAYERS. Club especializado en Commodore-Amstrad. Muchos programas. Gran experiencia. Zona de Almería. Interesados llamar al 254501 de Almería y preguntar por Angel. Angel D. Jiménez. S. Leonardo, 24 1D. 04004-ALMERIA.

CONTACTOS

- Desearía contactar con usuarios o bien clubs de C-64 en la zona de Gerona. Josep M. Rodríguez. Conill, 20 at.2. 17820- BANYOLES (GERONA) Tlf.: 574858.
- Intercambio todo tipo de información sobre AMIGA 500. Jose M. Campos Martinez. Antonio Palacios, 58 2A. 36400-PORRIÑO (PONTEVEDRA) Tlf.: 986-335809.
- Deseo contactar para intercambio de ideas, proyectos y programas, escribir a: Julio Espinosa Mercado. Francisco Hernandez Guerra, 8 7C. 35014-LAS PALMAS.
- Deseo contactar con amigos del C-64. Juan Martin Gril. Avda. Pueyrredon, 923 12A. 1032-BUENOS AIRES (CAPITAL FEDERAL) ARGENTINA. Tlf.: 962-2379.

pedidos

commodore WORLD

Para hacer tus pedidos, fotocopia esta página (o envíanos el pedido por carta) y marca lo que quieras con una cruz. Suma tú mismo el importe y envíanos un cheque o giro por el total.

NUMEROS ATRASADOS

20	21	22	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43
44	45	46	47	48	49	50	51	52	53	54
55	56	57	57	58	59	60	61			

(Señala con un circulo los números que quieras)

Precios de los ejemplares:
- Hasta el número 32 a ..... 300 ptas.
- Del 33 al 43 a ..... 375 ptas.
- Del 44 en adelante ..... 400 ptas.

Los números que no figuran se encuentran agotados.

- ☐ Número atrasado + disco del mismo número ..... 1.950 ptas.
- ☐ Oferta: 7 números atrasados + tapas de regalo ..... 2.345 ptas.
- ☐ Tapas de encuadernación (para 12 números) ..... 795 ptas.

EJEMPLARES ATRASADOS DE "CLUB COMMODORE" (Servicio de fotocopias)

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15

(Señala con un círculo los números que quieras)

- ☐ Ejemplar Club Commodore ..... 370 ptas.
- ☐ Oferta: Colección completa (16 números) ..... 3.100 ptas.

BIBLIOTECA COMMODORE WORLD

- ☐ Volumen 1: Cursillo de código máquina ..... 250 ptas.
- ☐ Volumen 2: Especial Utilidades ..... 500 ptas.
- ☐ Disco Especial Utilidades ..... 1.750 ptas.
- ☐ Oferta: Especial Utilidades + Disco ..... 1.990 ptas.

DISCOS DEL MES

Estos discos contienen todos los programas de la revista del mes correspondiente, incluyendo *(completos)* tanto los que se publican en varias partes como las "mejoras". Se suministra gratuitamente el programa "Datafile" (versión C-128) que contiene el "índice Commodore World", que se actualiza mes a mes.

14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44	45	46
47	48	49	50	51	52	53	54	55	56	57
58	59	60	61							

- ☐ Disco del mes ..... 1.750 ptas.
- ☐ Oferta: 5 discos del mes ..... 7.990 ptas.
- ☐ Suscripción un año (11 discos) + 11 revistas a partir del número ☐ ..... 17.500 ptas.

(Señala con un círculo los discos que deseas pedir)

PROGRAMOTECA COMMODORE WORLD

Estos discos incluyen instrucciones de funcionamiento para todos los programas que contienen:

- ☐ Superdisco Aplicaciones I (dos discos) ...... 1.990 ptas.
- ☐ Superdisco Aplicaciones II (dos discos) ..... 1.990 ptas.
- ☐ Superdisco Aplicaciones III (dos discos) ..... 1.990 ptas.
- ☐ Superdisco Aplicaciones IV (dos discos) ..... 1.990 ptas.
- ☐ Superdisco Juegos ..... 1.375 ptas.

¡¡ATENCION!! Discos y revistas

AMIGA

	Ptas.
☐ Amiga World 0, 1	500
☐ Oferta: Amiga World 0, 1 + disco especial	1.995
☐ Oferta: Todo sobre Amiga (28, 34, 35, 38 y 40 al 57)	4.500

Discos Amiga 1, 2, 3, 4 y Discos Especiales Amiga World 0, 1.

- ☐ Un disco ..... 1.700 ptas.
- ☐ Oferta: tres discos ..... 4.500 ptas.
- ☐ Oferta: cinco discos ..... 7.000 ptas.

No se admiten pedidos contrarreembolso. Enviar a: COMMODORE WORLD. Rafael Calvo,18 - 4.º B. 28010 MADRID.

Nombre y Apellidos ..... Teléfono .....

Dirección .....

Población ..... Provincia ..... C.P. ..... Modelo de ordenador .....

Importe del pedido ..... ptas. Forma de pago: ☐ Cheque ☐ Giro número ..... (Indicar pedido en el giro)

Tarjeta: ☐ VISA ☐ MASTERCARD N.º tarjeta. Fecha caducidad ..... (Los pedidos con tarjeta de crédito, sólo a partir de 3.000 ptas.)

Gastos de envío e IVA incluidos.

CLAVE PARA INTERPRETAR LOS LISTADOS

Todos los listados que se publican en Commodore World han sido cuidadosamente comprobados en el modelo correspondiente de los ordenadores Commodore. Para facilitar su edición y para mejorar la legibilidad por parte del usuario se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Dado que los ordenadores Commodore utilizan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos, así como movimientos del cursor, códigos de color, etc., por equivalencias entre corchetes que indican la secuencia de teclas que se debe pulsar para obtener dichos caracteres.

Las claves constan siempre de una letra o símbolo precedidos de las palabras COMM, SHIFT o CTRL; por ejemplo [COMM +] o [SHIFTA]. Esto indica que para obtener el gráfico hay que pulsar a la vez la tecla COMMODORE (la de abajo a la izquierda) o la tecla SHIFT (cualquiera de ellas) junto con la letra o símbolo correspondiente, en este ejemplo "+" o "A".

También puede aparecer un número indicando cuántas veces hay que repetir el carácter.

[7 CRSRR] equivale a siete cursores a la derecha y [3 SPC] a tres pulsaciones de la barra espaciadora.

```
1 REM "PERFECTO"                          .113
2 REM POR JAMES E. BORDEN, RUN EEUU .96
3 REM (C)1986 COMMODORE WORLD            .157
4 :                                        .236
5 POKE56,PEEK(56)-1:POKE52,PEEK(56) .119
6 CLR:PG=PEEK(56):ML=PG*256+60           .232
7 :                                        .239
8 P=ML:L=24                                .216
9 S=0:FORI=0TO6:READA:IFA=-1THEN16   .59
10 IFA<0ORA>255THEN14                    .146
11 POKEP+I,A:S=S+A:NEXT                  .81
12 READSC:IFS<>SCTHEN14                  .250
13 L=L+1:P=P+7:GOTO9                     .97
14 PRINT"ERROR EN DATAS LINEA";L:EN .60
D
15 :                                       .247
16 POKEML+4,PG:POKEML+10,PG              .60
17 POKEML+16,PG:POKEML+20,PG             .221
18 POKEML+32,PG:POKEML+38,PG             .110
19 POKEML+141,PG                          .97
20 SYSML:PRINT"[CRSRD] [WHT]CORRECT .98
OR ACTIVADO
21 PRINT" SYS"ML"=CONECTAR               .127
22 PRINT" SYS"ML+30"=DESCONECTAR[CO .122
MM7]
23 :                                       .255
24 DATA173,5,3,201,3,208,1,594           .22
25 DATA96,141,105,3,173,4,3,525          .181
26 DATA141,104,3,162,103,160,3,676      .214
27 DATA142,4,3,140,5,3,96,393            .177
28 DATA234,234,173,104,3,141,4,893      .96
29 DATA3,173,105,3,141,5,3,433           .177
30 DATA96,32,124,165,132,11,162,722 .18
31 DATA0,142,240,3,142,241,3,771        .87
32 DATA189,0,2,240,51,201,32,715        .166
33 DATA208,4,164,212,240,40,201,106 .177
9
34 DATA34,208,8,72,165,212,73,772       .146
35 DATA1,133,212,104,72,238,241,100 .237
1
36 DATA3,173,241,3,41,7,168,636         .142
37 DATA104,24,72,24,104,16,1,345        .225
38 DATA56,42,136,16,246,109,240,845 .238
39 DATA3,141,240,3,232,208,200,1027 .123
40 DATA173,240,3,24,101,20,24,585       .72
41 DATA101,21,141,240,3,169,42,717      .49
42 DATA32,210,255,169,0,174,240,108 .170
0
43 DATA3,32,205,189,162,4,189,784       .83
44 DATA211,3,32,210,255,202,16,929      .214
45 DATA247,164,11,96,145,13,32,708      .87
46 DATA32,0,0,0,0,0,0,32,-1              .146
```

CLAVE	EQUIVALENCIA
CRSRD	CURSOR ABAJO (SIN SHIFT)
CRSRU	CURSOR ARRIBA (CON SHIFT)
CRSRR	CURSOR DERECHA (SIN SHIFT)
CRSRL	CURSOR IZQUIERDA (CON SHIFT)
HOME	CLR/HOME SIN SHIFT
CLR	CLR/HOME CON SHIFT
SPC	BARRA ESPACIADORA
DEL	INST/DEL Y SHIFT + INST/DEL
INST	INST/DEL CON SHIFT
BLK A YEL	COLORES: CONTROL + NUMERO
RVS ON	CONTROL + 9
RVS OFF	CONTROL + 0
F1 A F8	TECLAS DE FUNCION
FLCH ARRIBA	FLECHA ARRIBA
FLCH IZQ	FLECHA A LA IZQUIERDA
PI	PI (FLECHA ARRIBA CON SHIFT)
LIBRA	LIBRA
PARA C-128	
BELL	CONTROL + G
TAB	TAB O CONTROL + I
LFEED	LINE FEED O CONTROL + J

Cómo utilizar la suma de control

Todos los listados para C-64 o C-128 que aparecen en la revista llevan una suma de control para que no te equivoques a la hora de teclearlos. Para poder utilizar esta suma de control tendrás que teclear el programa que aparece aquí listado. Se llama "Perfecto". Cuando lo hayas tecleado, compruébalo y sálvalo en disco o cinta.

Cada vez que vayas a teclear uno de los programas que aparecen en Commodore World debes cargar antes el programa corrector. Apunta los dos valores SYS que aparezcan en la pantalla, pues te servirán para conectar o desconectar el programa cuando quieras.

Verás que a la derecha de todos los listados aparece un punto seguido de un número. Eso no debes teclearlo, pues en tal caso el programa no te funcionaría. Comienza a teclear el listado normalmente. La única diferencia que notarás es que al pulsar RETURN aparecerá un asterisco seguido de un número de una, dos o tres cifras debajo del cursor. Es la suma de control. Compárala con el número que aparece en la parte derecha del listado. Si es el mismo, puedes seguir tecleando, pero si es diferente deberás buscar errores en la línea que acabes de introducir. Observa sobre todo los siguientes puntos:

- Los espacios sólo se tienen en cuenta si van entre comillas. Los demás los puedes omitir. Si tienes problemas con alguna línea tecléala tal y como aparece en el listado, ¡teniendo en cuenta las claves, por supuesto!
- Los comandos Basic se pueden abreviar, de modo que puedes poner ? en vez de PRINT o P SHIFT O en vez de POKE.
- También se tiene en cuenta el número de línea. Si por error introduces la línea 100 en vez de la 1000, por ejemplo, tendrás que teclear nuevamente la línea 100 (que se habrá borrado) y a continuación la 1000.
- Si quieres modificar alguna línea a tu gusto, obtén primero la suma de control correcta y luego modifícala.

Si por alguna razón no consigues la misma suma de control que aparece en el listado, prueba a borrar la pantalla y teclear la línea entera de nuevo. Un artículo completo sobre el funcionamiento de este programa apareció en el número 23 de Commodore World.

Super Discos Aplicaciones IV

UNA VEZ MAS HEMOS PREPARADO UN CONJUNTO DE PROGRAMAS ESPECIALMENTE PENSADOS PARA LOS USUARIOS DE COMMODORE 64 Y 128



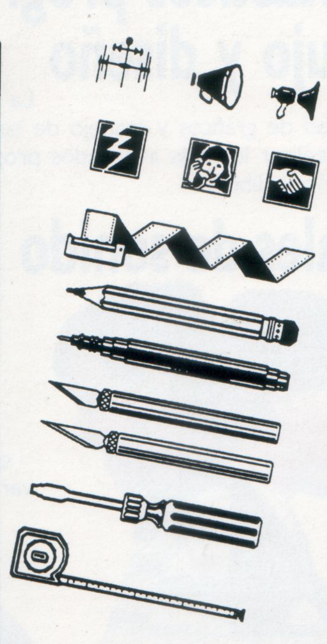
En estos dos discos APLICACIONES IV se encuentran reunidos programas totalmente nuevos (no publicados anteriormente en Commodore World o en otros discos) y algunos programas escogidos de entre los mejores publicados en la revista. Desde que se lanzó el número III, han aparecido algunas utilidades interesantes que hemos considerado oportuno recopilar para todos los usuarios commodorianos.

PROGRAMAS PARA C-64

- **GRAPHMAKER.** Producir e imprimir gráficos tridimensionales con gran sencillez en su C-64 o C-128, en 40 ú 80 columnas.
- **LABEL BASE.** Un completo editor de etiquetas para sus tareas de correos. Podrá crear etiquetas de sus direcciones de bases de datos.
- **RUN PAINT.** El programa artístico que no puede faltar en cualquier programoteca de un commodoriano. Es completísimo y muy bueno, mejor que algunos programas comerciales.
- **RUN TERM.** Por fin podemos presentar un paquete de comunicaciones perfectamente probado en C-64 y C-128. Es sencillo de manejo y superpotente.
- **MAQUINA DE SONIDO 5.0.** La última versión del programa de música más perfeccionado en Commodore World.

(A partir del siguiente, programas publicados en Commodore World)

- **COMPACTOR 64.** Un programa de utilidad que facilita las tareas de los programadores en Basic. Haciendo honor a su nombre, es muy compacto.
- **HELP SYSTEM.** Programa residente en memoria. Sistema de ayuda con calculadora, reloj y otras utilidades.
- **PAGE LISTER.** Listador página a página para permitir comprobar o estudiar mejor los listados de tu C-64.
- **PIANOLA 64.** Un programa de música que, siendo sencillo, posibilita la edición de música por interrupciones.
- **SIZZLE.** Este turbo de disco sirve para acelerar la carga y grabación de programas en modo C-64.
- **SUPERFICIES 3D.** Una maravilla de la generación gráfica tridimensional. Además de útil, se puede aprender sobre programación de gráficos.



Se puede realizar hoy mismo el pedido de estos discos mediante el envío del boletín a:

PROGRAMAS PARA C-128

- **EDITOR MUSICAL 128.** Como su nombre indica se trata de un programa especialmente diseñado aprovechando las características del C-128 para la edición musical.
- Versiones especiales de RUN PAINT y RUN TERM. Para modo 128 también GRAPH MAKER y LABEL BASE.

Además, todos los programas para C-64 funcionan perfectamente en el modo 64 del Commodore 128.

LOS DOS DISCOS AL INCREIBLE PRECIO DE 1.990 ptas.

BOLETIN DE PEDIDO – Especial Aplicaciones IV

Nombre y Apellidos ....

Dirección ....

Población .... C.P. .... Provincia .... Teléfono ....

Forma de Pago: Sólo cheque o giro. No se sirven pedidos contrarreembolso. Gastos de envío e IVA incluidos.

☐ Incluyo cheque por 1.990 ptas.

☐ Envío giro número .... por 1.990 ptas.